

CMS "is.anonyme"

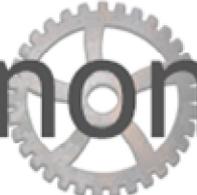
Développement d'un petit CMS dédié à l'IT

Web ; Intranet ; CMS

25.10.2023

Version 3

Auteur : fond-fonfon.ch

Anonyme SA

Site Web "is.anonyme"

Mots clefs: Site Web; Intranet

Index

Index	2
Avertissement	13
Droits de copie	13
Développement.....	13
Versions	13
Conventions	14
Rédaction.....	14
Introduction	15
Source d'information	16
Note	16
Description et composants du système	17
Serveur.....	17
Paramètres de la machine virtuelle	17
Système d'exploitation actuel.....	17
Principaux logiciels installés sur le serveur "Linux Ubuntu".....	18
Modules	18
Langages de développement employés.....	18
Site "is.anonyme "	19
Architecture du site web.....	19
Arborescence	20
Répertoires racines sur le serveur	20
Répertoires du site web	20
Sécurité des répertoires.....	21
Modules complémentaires	22
PHPGrid	22
phpGridHandle	22
LDAP	24
Mode console.....	24
Mode "Web"	24
Base de données	25
Les tables	25
Préfixes de tables.....	25
Liste des tables	25

Liste des vues	27
Procédures stockées.....	27
Développement du code "PHP"	28
Page "index.php"	28
Page "authentication.php"	29
Page "exec(.table).php"	30
Menu principal	31
Table "_menu".....	31
Description des champs de la table "_menu"	32
Lien interne de type "uri"	33
Classes PHP	34
Classe "authentication.class.php"	35
Méthodes de classe	35
I. public function authenticate (\$usr, \$pwd, \$db_hdl, \$uri = "").....	35
Variables	35
Valeur retournée par la fonction	35
II. public function isAuthenticated (\$db_hdl).....	35
Variables	35
Valeur retournée par la fonction	35
III. public function isPasswordTemporary(\$db_hdl)	36
Variables	36
Valeur retournée par la fonction	36
IV. public function checkPassword(\$db_hdl, \$usr, \$pwd)	36
Variables	36
Valeur retournée par la fonction	36
V. public function showFormAuthenticate (\$uri = "")	36
Variables	36
Valeur retournée par la fonction	36
VI. public function showFormPwdGen (\$db_hdl, \$html_hdl, \$uri = "")	37
Variables	37
Valeur retournée par la fonction	37
VII. public function showFormInfo (\$aUser, \$encCode, \$url, \$msg = "", \$clr = "").....	37
Variables	37
Valeur retournée par la fonction	37
I. private function writeToLogTableFromAuth(\$db_hdl, \$table, \$type = "information", \$op = "", \$side = "", \$service = "", \$action = "", \$log = "").....	38
Variables	38
Valeur retournée par la fonction	39
Classe "html.class.php".....	40
Propriétés de classe	40
Méthodes de classe	40
I. public function showOpenHTMLEnv(\$site, \$dirCSS = "", \$dirPIC = "", \$dirSCR = "")	40

Variables	40
Valeur retournée par la fonction	40
II. public function showTopMargin()	41
Valeur retournée par la fonction	41
III. public function showOpenMainBackground()	41
Valeur retournée par la fonction	41
IV. public function showCloseMainBackground()	41
Valeur retournée par la fonction	41
V. public function showCloseHTMLEnv	41
Valeur retournée par la fonction	41
VI. public function generateOptionList(\$listID, \$datalistID, \$anArray, \$subKeys = "", \$subKeysSep = "", \$sep = "", \$js = "", \$class = "clDataList")	41
Variables	41
Valeur retournée par la fonction	43
VII. public function convertToAsciiCodeForHTML(\$s)	43
Variable	43
Valeur retournée par la fonction	43
VIII. public function getPicturesFromAsciiCode(\$s, \$params = "")	43
Variables	43
I. private function curlRequest(\$url)	43
Variable	43
Valeur retournée par la fonction	43
II. private function getBinaryData(\$file, \$mime)	44
Variables	44
Valeur retournée par la fonction	44
Classe "mail.class.php"	45
Propriétés de classe	45
Méthodes de classe	48
I. public function __construct(\$db_h = Null)	48
Variable	48
Valeur retournée par la fonction	48
II. public function prepareEMailMessage()	48
Valeur retournée par la fonction	48
III. public function prepareEMailMessageForAGGroup()	48
Valeur retournée par la fonction	48
IV. public function prepareEMailMessageForPwdExp()	48
Valeur retournée par la fonction	48
V. public function prepareEMailMessageForPwdReset()	48
Valeur retournée par la fonction	48
Fonctions liées à des groupe particuliers.....	49
VI. public function getAGTasksForToUsers(\$db_hdl)	49
Variable	49

Valeur retournée par la fonction	49
VII. public function prepareEMailMessageForcalEvent(\$count, \$method = 'REQUEST')	49
Variables	49
Valeur retournée par la fonction	49
Fonctions liées à l'envoi de courriels	50
VIII. public function sendMail(\$messageType = "", \$message = "")	50
Variables	50
Valeur retournée par la fonction	50
IX. public function sendMailReminders(\$messageType = "")	51
Variable	51
Valeur retournée par la fonction	51
Fonctions utilitaires	52
X. public function getMailAddress(\$user)	52
Variable	52
Valeur retournée par la fonction	52
I. private function getTableDatas()	52
<i>Valeur retournée par la fonction</i>	52
Classe "menu.class.php"	53
Méthodes de classe	53
I. public function showBeginTopMenuSection(\$section, \$rowAlign, \$rowSection)	53
Variables	53
Valeur retournée par la fonction	53
II. public function showEndTopMenuSection()	53
Valeur retournée par la fonction	53
III. public function showBeginSubMenuSection()	53
Valeur retournée par la fonction	53
IV. public function showEndSubMenuSection	53
Valeur retournée par la fonction	53
V. public function showTopMenuItem()	54
Valeur retournée par la fonction	54
VI. public function showSubMenuItem()	54
Valeur retournée par la fonction	54
VII. public function showMenuInter(\$count, \$result, \$align = "")	55
Variables	55
Valeur retournée par la fonction	55
I. private function setLink(\$row)	55
Variable	55
Valeur retournée par la fonction	55
Classe "mysql.class.php"	56
Propriétés de classe	56
Méthodes de classe	56
I. public function __construct(\$_MYSQL)	56

Variable	56
Valeur retournée par la fonction	56
II. public function getPfxTables(\$tablePrefix)	56
Variable	56
Valeur retournée par la fonction	56
III. public function query(\$query)	56
Variable	56
Valeur retournée par la fonction	56
IV. public function fetch_assoc(\$query)	57
Variable	57
Valeur retournée par la fonction	57
V. public function saveTableData(\$tableName, \$aData = array(), \$condition = "")	57
Variables	57
Valeur retournée par la fonction	57
VI. public function resetTable(\$tableName)	57
Variable	57
Valeur retournée par la fonction	57
VII. public function resetFKTable(\$tableName).....	58
Variable	58
Valeur retournée par la fonction	58
VIII. public function modSafeUpdateMode(\$mode).....	58
Variable	58
Valeur retournée par la fonction	58
IX. public function sql_disconnect().....	58
Valeur retournée par la fonction	58
I. private function show_errors(\$err)	58
Variable	58
Valeur retournée par la fonction	58
Classe "mysqlisanonyme.class.php"	59
Méthodes de classe	59
I. public function writeToLogTable (\$table, \$type = "information", \$op = "", \$side = "", \$service = "", \$action = "", \$log = "").....	59
Variables	59
Valeur retournée par la fonction	60
II. public function addDBColumnToTableVsLDAP(\$table, \$col, \$type, \$default = "NULL").....	60
Variables	60
Valeur retournée par la fonction	60
III. public function deleteDBColumFromTableVsLDAP(\$table, \$col).....	61
Variables	61
Valeur retournée par la fonction	61
IV. public function getDataFrom_grids_definersTableView(\$table, \$field = "").....	61
Variables	61

Valeur retournée par la fonction	61
V. public function colNameExistsIn_grids_definers(\$bench, \$field)	61
Variables	61
Valeur retournée par la fonction	61
VI. public function addColNameInRowTo_grids_definers(\$bench,\$colName)	62
Variables	62
Valeur retournée par la fonction	62
VII. public function deleteColAsRowFrom_grids_definers(\$row)	62
Variable	62
Valeur retournée par la fonction	62
VIII. public function getDataFrom_acr_TableView(\$table)	62
Variable	62
Valeur retournée par la fonction	62
IX. public function getUserGroupsACR(\$usr).....	63
Variable	63
Valeur retournée par la fonction	63
X. public function getEffectiveACRMenu(\$usr)	63
Variable	63
Valeur retournée par la fonction	63
XI. public function setGridACR(\$usr, \$bench).....	63
Variables	63
Valeur retournée par la fonction	63
XII. public function getUsersFromDB().....	64
Valeur retournée par la fonction	64
XIII. public function getUserFromDB(\$usr).....	64
Variable	64
Valeur retournée par la fonction	64
XIV. public function setNewUserPasswordInDB(\$usr, \$pwd, \$isTemp, \$cost=8)	64
Variables	64
Valeur retournée par la fonction	64
XV. public function verifyNewUserPasswordInDB(\$usr, \$pwd).....	65
Variables	65
Valeur retournée par la fonction	65
XVI. public function getTableValueVsID(\$table, \$field, \$id)	65
Variables	65
Valeur retournée par la fonction	65
XVII. public function getTableValueVsFieldValue(\$table, \$getFields, \$field, \$fieldVal, \$isStr = false)....	66
Variables	66
Valeur retournée par la fonction	66
XVIII. public function getAllTableValuesVsID(\$table, \$id)	66
Variables	66
Valeur retournée par la fonction	66

XIX. public function setTableValueVsID(\$table, \$field, \$id, \$val)	67
Variables	67
Valeur retournée par la fonction	67
XX. public function getToDoInfosFromAGTasksVsID(\$userKeys,\$idAGT).....	67
Variables	67
Valeur retournée par la fonction	67
XXI. public function getDocumentsTree().....	67
I. private function getGridACRvsUser(\$usr,\$bench).....	67
Variables	67
Valeur retournée par la fonction	67
Classe "phpgrid.class.php"	68
Propriétés de classe	68
Méthodes de classe.....	70
I. public function __construct(\$sqlTable = Null, \$viewType = Null, \$db_h = Null, \$pg_h = Null)	70
Variables	70
Valeur retournée par la fonction	70
II. public function openHtmlEnvironment(\$supHtml = "").....	71
Variable	71
Valeur retournée par la fonction	71
III. public function closeHtmlEnvironment().....	71
Valeur retournée par la fonction	71
IV. public function setHtmlJQGridStyle().....	72
Valeur retournée par la fonction	72
V. public function setHtmlDivWidthConstraint().....	72
Valeur retournée par la fonction	72
VI. public function setHtmlToggleToolbar()	72
Valeur retournée par la fonction	72
VII. public function setExportOptions().....	72
Valeur retournée par la fonction	72
VIII. public function setGridSizeFromMenuViewStyle().....	72
Valeur retournée par la fonction	72
IX. public function setColWidth(\$col, \$width)	73
Variables	73
Valeur retournée par la fonction	73
X. public function setColsWidths()	73
Valeur retournée par la fonction	73
XI. public function prepareStatements().....	73
Valeur retournée par la fonction	73
XII. public function executeStatements().....	73
Valeur retournée par la fonction	73
I. private function checkUsersVsGroup(\$groups)	74
Variable	74

Valeur retournée par la fonction	74
II. private function getValuesFromDefinersForMandatoryFields(\$table)	74
Variable	74
Valeur retournée par la fonction	74
III. private function getValuesFromDefinersForSelectFields(\$table, \$field)	74
Variables	74
Valeur retournée par la fonction	74
Fonctions liées à des tables particulières	75
I. private function getAssetsInventoryNumbersForSelect()	75
Valeur retournée par la fonction	75
II. private function getUsersFNForSelect()	75
Valeur retournée par la fonction	75
III. private function getUsersUNForSelect()	75
Valeur retournée par la fonction	75
IV. private function getUsersUNVsGroupForSelect(\$group = "")	75
Valeur retournée par la fonction	75
V. private function getGroupsNForSelect()	75
Valeur retournée par la fonction	75
VI. private function getMaterialsForSelect()	76
Valeur retournée par la fonction	76
VII. private function getAddressesDataForSelect(\$field)	76
Valeur retournée par la fonction	76
VIII. private function getCategoriesForSelect(\$table, \$field)	76
Variables	76
Valeur retournée par la fonction	76
Fonctions liées à des pseudo-événements "jqGrid"	77
I. public function onBeforeShowForm_SetDateModificationRecord()	77
Evènement	77
Valeur retournée par la fonction	77
II. public function onBeforeShowForm_SetUserModificationRecord(\$user)	77
Evènement	77
Variable	77
Valeur retournée par la fonction	77
III. public function onBeforeShowForm_SetUserCreationRecord(\$user)	77
Evènement	77
Variable	77
Valeur retournée par la fonction	77
IV. public function onBeforeShowForm_DeleteRecord()	78
Evènement	78
Valeur retournée par la fonction	78
V. public function onEditAfterShowForm(\$col)	78
Evènement	78

Variable	78
Valeur retournée par la fonction	78
VI. public function onAfterDeleteRecord(\$section, \$url)	78
Evènement	78
Variables	78
Valeur retournée par la fonction	78
VII. public function onBeforeSubmitRecord()	79
Evènement	79
Valeur retournée par la fonction	79
VIII. public function onAfterSubmitRecordWithMail(\$section, \$url)	79
Evènement	79
Variables	79
Valeur retournée par la fonction	79
IX. public function onAfterSubmitRecordNew(\$section, \$url)	79
Evènement	79
Variables	79
Valeur retournée par la fonction	79
X. public function onBeforeScriptEnd_MultiExport(\$path, \$csv = 0, \$excel = 0, \$html = 0, \$pdf = 0)	80
Evènement	80
Variables	80
Valeur retournée par la fonction	80
XI. public function onAfterEditShowForm_PermitAddWithReadOnlyFields(\$aFields)	81
Evènement	81
Variable	81
Valeur retournée par la fonction	81
Fonctions "utilitaires" de la classe	81
XII. public function getMenuLabelFromSection(\$section)	81
Variable	81
Valeur retournée par la fonction	81
XIII. public function manageTableRowAttachment(\$id, \$op, \$absURL, \$absURI)	82
Variables	82
Valeur retournée par la fonction	82
XIV. public function getTableRowAttachment(\$id)	82
Variable	82
Valeur retournée par la fonction	82
XV. public function getTableRowAttachmentExt(\$id)	83
Variable	83
Valeur retournée par la fonction	83
XVI. public function setTableRowAttachmentLink(\$id, \$file, \$absURL, \$absURI)	83
Variables	83
Valeur retournée par la fonction	83
XVII. public function setTableRowAttachmentIcon(\$id,\$ext)	83

Variables	83
Valeur retournée par la fonction	83
XVIII. public function unsetTableRowAttachmentIcon(\$id)	84
Variable	84
Valeur retournée par la fonction	84
Table "_grids_definers"	84
Description des champs de la table "_grids_definers"	84
Classe "security.class.php"	86
Propriétés de classe	86
Méthodes de classe	88
I. public function setEncoding(\$str, \$newEncoding)	88
Variables	88
Valeur retournée par la fonction	88
II. public function blacklistFilter(\$str)	88
Variable	88
Valeur retournée par la fonction	88
III. public function whitelistFilter(\$str, \$whiteFilterPattern = "", \$errCode = 0, \$min = 8, \$max= 68, \$urlRedirect = "")	89
Variables	89
Valeur retournée par la fonction	90
IV. public function riseSecError (\$eCode)	91
Variable	91
Valeur retournée par la fonction	91
V. public function riseSecErrorAfterHeaders (\$eCode)	91
Variable	91
Valeur retournée par la fonction	91
VI. public function checkBrowserAndIP(\$ipTVF, \$UATVF, \$errCode = 999)	91
Variables	91
Valeur retournée par la fonction	91
VII. public function execlFFileExists (\$pathToFile)	92
Variable	92
Valeur retournée par la fonction	92
VIII. public function decryptAsJSSimpleDecrypt(\$str)	92
Variable	92
Valeur retournée par la fonction	92
Classe "session.class.php"	93
Méthodes de classe	93
I. public function preventHijacking()	93
Valeur retournée par la fonction	93
Annexes	94
Annexe I. Affichage du tableau "d_assets" selon la vue "standard"	95
Principe	95

Schéma bloc.....	97
Code "typique" d'une page de section pour une utilisation avec "PHPGrid".....	98
Légende du code PHP.....	98
Code.....	98
Architecture de la table "d_assets".....	101
Paramètres d'affichage de la table "d_assets" présents dans la table "_grids_definers".....	102
Page résultat.....	103
Formulaire de saisie associé au tableau.....	104
Table des illustrations.....	105

Avertissement

Droits de copie

1. Le site nommé ci-après "is.anonyme", est un système Intranet à vocations multiples, de type "plateforme multiusage" ou "Content Management System".
2. L'ensemble du code développé pour le système "is.anonyme" est propriété de fond-fonfon.ch, à l'exclusion des mentions faites au point "4" de la présente section.
3. Une autorisation d'usage peut être obtenue auprès de "fond-fonfon.ch", par l'entremise de l'envoi d'un courriel à l'adresse "info@fond-fonfon.ch". Cela inclut la reproduction, même partielle, du présent contenu de ce document ou du code qui y est produit.
4. Ne sont pas concernés par le présent droit de copie, le ou les module(s) et/ou fichier(s) suivants :
 - Le module "PHPGrid", qui est un produit commercial, acquis auprès de "Kayson Group Inc." (<https://phpgrid.com>), dont le code source est soumis à la législation des Etats-Unis, relativement au droit de copie, ainsi qu'aux traités internationaux y afférant. Se référer aux dispositions prévues par "Kayson Group Inc." quant à son usage.

Développement

1. Le code logiciel développé dans ce document n'a pas vocation à être parfait. Il est donc susceptible de contenir des erreurs, que ce soit au niveau architectural ou au niveau "sémantique" et a donc vocation à être optimisé.
2. Néanmoins, il a été développé, testé et mis en production et sa fonctionnalité a été démontrée.
3. Le code développé dans ce document se réfère aux versions respectives de "PHPGrid v7.5" et "PHP v7.4.x".

Versions

- Attention, dès la version 8.1.0 de PHP, le comportement des variables globales change, donnant lieu à des erreurs lors de l'interprétation du code (cf. [\\$GLOBALS sur le site php.net](#)). Il convient donc de respecter les diverses versions des logiciels mentionnés tout au long de ce document.
- Ce cas de figure est également avéré pour la version du produit "PHPGrid", vis-à-vis de la version de "PHP" au-delà de "7.4".
- De tels aspects peuvent se produire dans d'autres cas, non répertoriés.

Conventions

- Enregistrement : Le contenu d'une "ligne" complète d'une table MySQL
- Liste : désigne une table MySQL préfixée "d_" pour "données"
- Section : Désigne une page du site Web
- Table, Tableau : désigne une table MySQL
- Caractérisation des colonnes dans les tables MySQL :

PK	Clef Principale indexée (Primary Key)
NN	Ne peut être égal à "Null" (Not Null)
UQ	Index Unique (UniQue)
B	Colonne binaire (Binary column)
UN	Non signé (UNsigned)
ZF	Rempli les colonnes avec un "0" si celle-ci est numérique (Zero Fill)
AI	Incrément automatique (Auto Increment)
G	Colonne générée (Generated column)

- Vue : Une représentation particulière d'une table MySQL dans le contexte du site Web, selon les besoins.
- "AG" ou "ag" ou "AGGroup" : Sous forme d'insertion dans des titres de fonctions ou individuellement dans le texte, désigne un groupe de travail fictif (entendez "équipe de personnes"), dénommé "Anonymous Group" ou "Groupe Anonyme".

Rédaction

- Malgré tout le soin apporté à la production du présent document, certaines "coquilles" sont susceptibles de persister.

Introduction

Le système Intranet "is.anonyme" a été développé en premier lieu pour l'usage du Service informatique de l'entreprise "Anonyme SA".

Il résulte d'une évolution d'un système précédent : "Microsoft SharePoint Foundation version 2010", alors utilisé par le Service informatique pour divers usages (listings de support, d'inventaires, etc.).

Pour succéder à Microsoft SharePoint, devenu payant depuis 2015, plusieurs solutions telles que des plateformes "CMS" (Content Management System), "Wordpress" ou "Joomla" par exemple, ont été étudiées.

Toutefois, ces produits ne constituaient pas une solution satisfaisante en soi pour remplacer SharePoint, qui intégrait un ensemble de fonctionnalités propriétaires, telles que la gestion de listes.

A contrario, les "CMS" testés demandaient trop d'adjonctions annexes, sous forme de modules ou d'extensions, qui n'étaient soit pas ou peu utilisés, ou encore fastidieux quant à leur gestion. En sus, chaque extension ou module représentait un investissement que ce soit en termes financier ou de temps.

Également, dans le cas d'un système tel que "Wordpress" (ou d'autres), se posaient également des questions quant à leur sécurité, du fait de leur modularité.

Il a donc été décidé de cibler et rationaliser le coût au plus proche de ce qui était souhaité et nécessaire et s'affranchir de ces problématiques.

C'est pourquoi, le Service informatique a entamé le développement d'une plateforme bâtie autour d'un produit commercial unique "PHPGrid". Ce dernier permet entre autres choses, le développement de listes sous forme de tableaux, en langage PHP et est totalement personnalisable.

Il ne nécessite pas d'adjonctions annexes et reste relativement simple d'utilisation, même si ses fonctionnalités sont perfectibles. Il est en outre relativement bon marché.

Également, cela a permis une liberté de développement que n'offraient pas des plateformes du type "Wordpress", soumises à un cadre architectural restrictif et demandant un investissement trop important en temps, pour entamer des développements propriétaires.

Une solution a donc été développée sur le moyen/long terme, et a gagné en importance, puisque des départements hors Service informatique, y ont également vu un intérêt.

Source d'information

Pour établir un fonctionnement pérenne, il est nécessaire de prévoir une ou plusieurs sources de données fiables.

Etant attendu que la finalité était de pouvoir au moins contenir une gestion d'actifs, un système de support (Service desk), ainsi qu'une base de prêt et de fournitures et une gestion des licences logicielles, il était nécessaire de pouvoir disposer d'au moins trois types de données, soit :

- Un registre complet des utilisateurs
- Un registre et des actifs
- Un registre de groupes
- Eventuellement un registre des unités organisationnelles

Une source d'information idéalement unique doit donc être privilégiée.

Un moyen permettant d'y parvenir de façon fiable est constitué par "Microsoft Active Directory", utilisé dans l'entreprise comme fondement de l'identité informatique des employés et d'une partie des actifs.

Active Directory permet en effet de caractériser des objets, utilisateurs ou actifs, en leur attribuant des droits et des attributs divers.

Par le biais du protocole "LDAP", la plateforme va donc pouvoir se baser sur un référentiel fiable de données, qui seront ensuite utilisées au sein des différentes applications développées sur la plateforme.

Pour réaliser cela, un module spécifique dédié "LDAP" a été développé en interne pour remplir la fonction de cette collecte d'information.

Note

Etant donné la complexité du site web, "l'Annexe I" du présent document propose un descriptif de son fonctionnement "par l'exemple". Le lecteur pourra s'y référer de façon à illustrer les propos tenus tout au long de ce document.

Description et composants du système

Cette section décrit l'infrastructure nécessaire au fonctionnement de la plateforme web.

Serveur

Machine virtuelle montée sur Microsoft Windows Server 2016 Hyper-V Cluster

Paramètres de la machine virtuelle

Point de configuration	Donnée
Mémoire RAM	8192 MB
Processeurs virtuels	2
Disque dur	256 GB
Réseau	Carte réseau sur sous-réseau serveurs
Nom	is.anonyme
Points de contrôle	Activés
Volume d'installation	C:\Clusterstorage\Volume2\is.anonyme
Démarrage	Démarrage automatique
Démarrage sécurisé	Activé (Microsoft UEFI Certificate Authority)

Système d'exploitation actuel

Linux Ubuntu 20.04.6 LTS

Se référer au document "2023-Serveur_Web_interne is.anonymme-Installation_du_systeme-v4.pdf", disponible sur le site "fond-fonfon.ch" pour la procédure propre à l'installation du système.

Principaux logiciels installés sur le serveur "Linux Ubuntu"

Logiciel	Version	Fonction
apache2	2.4.41	Serveur Web
php7.4	7.4.3	Plateforme de développement
php7.4- bz2; cgi; cli; common; curl; dev; fpm; gd; gmp; imap; intl; json; ldap; mbstring; mysql; opcache; pgsql; phpdbg; readline; sqlite3; sybase ; tidy; xml; xmlrpc; zip;		Extensions de la plateforme PHP
mysql-server	8.0.32	Serveur de base de données
phpmyadmin	4.9.5deb2	Gestion de la base de données
postfix	3.4.13	Serveur de messagerie
Vsftpd	3.0.3	Serveur FTP
zip	3.0	Outil de compression
(MySQL Workbench)	(8.0.23)	(Outil de gestion de base de données pour Windows)

Modules

Module	Version	Fonction
PHPGrid	7.5	Module PHP pour la construction/diffusion de tables/listes
LDAP	1	Module PHP de gestion LDAP

Langages de développement employés

Langages
CSS
HTML
PHP
JavaScript
SQL (MySQL)

Site "is.anonyme "

Architecture du site web

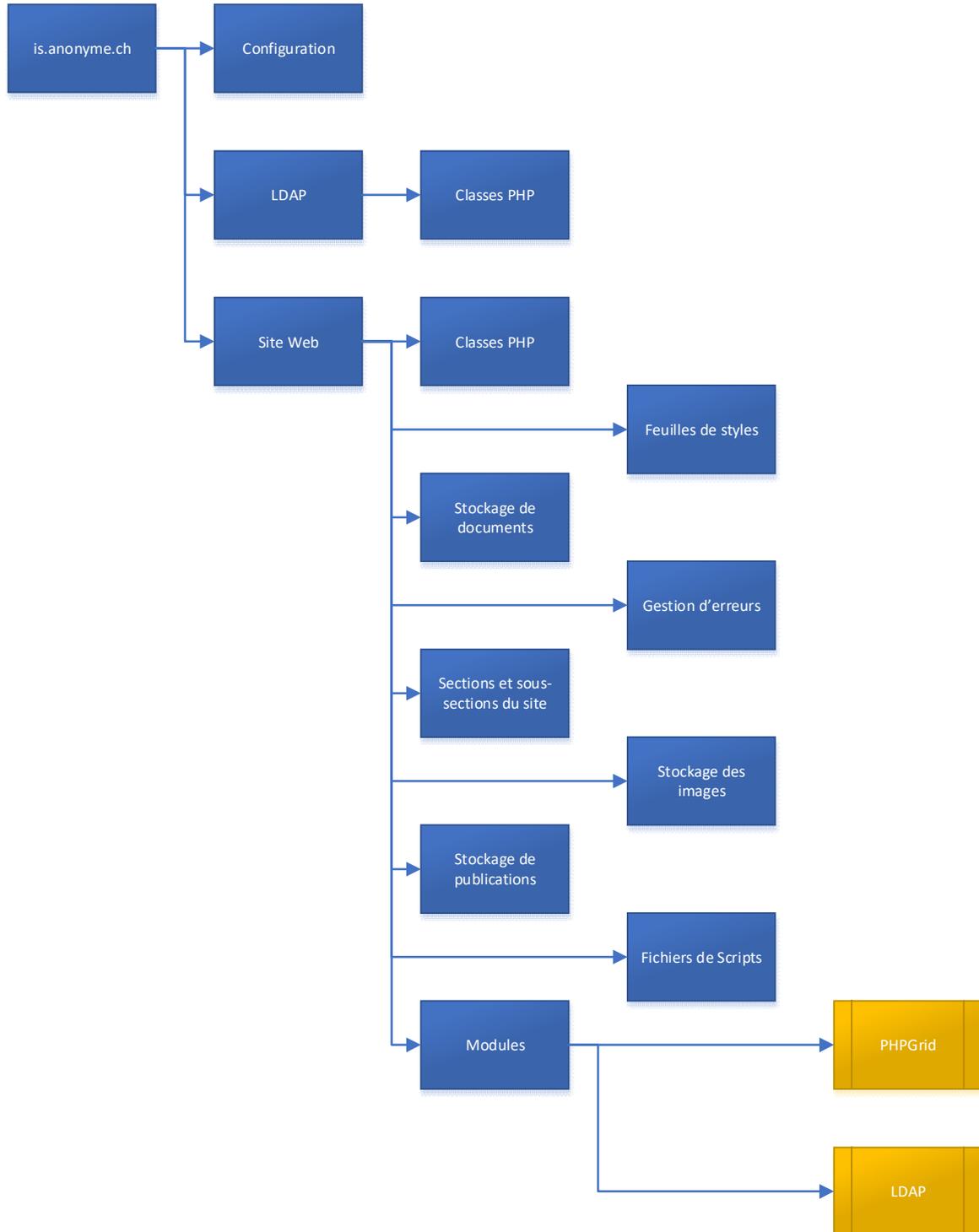


Figure 1: Structure du site web

Arborescence

Répertoires racines sur le serveur

Répertoire racine Apache pour servir les sites web : /var/www/vhosts

Répertoire racine du site "is.anonyme" : /var/www/vhosts/isanonyme/

Répertoires du site web

▼ [icône dossier]	Répertoire racine
▼ [icône dossier] boot	Base LDAP pour exécution en mode console
[icône dossier] class	Répertoire de classes PHP pour LDAP
[icône dossier] cfg	Données de configuration du site web
▼ [icône dossier] httpdocs	Répertoire racine du site web
[icône dossier] class	Stockage des classes PHP
[icône dossier] css	Stockage des feuilles de style
[icône dossier] docs	Stockage de documents
[icône dossier] errors	Stockage de la gestion d'erreurs
▼ [icône dossier] inc	Stockage des sections du site
[icône dossier] sections	Stockage des sous-sections du site (pages du site)
▼ [icône dossier] pictures	Stockage des images
[icône dossier] chars	Stockage des images représentant des caractères alphanumériques
[icône dossier] icons	Stockage des icônes
▼ [icône dossier] plugins	Stockage des modules
[icône dossier] ldap	Module "LDAP"
> [icône dossier] phpgrid	Module "PHPGrid"
[icône dossier] pub	Stockage de documents publiés
[icône dossier] scripts	Stockage des scripts

Figure 2: Structure des répertoires du site Web

Sécurité des répertoires

<ul style="list-style-type: none"> ▼  [redacted] ▼  boot  class  cfg ▼  httpdocs <ul style="list-style-type: none">  class  css  docs  errors ▼  inc <ul style="list-style-type: none">  sections ▼  pictures <ul style="list-style-type: none">  chars  icons ▼  plugins <ul style="list-style-type: none">  ldap >  phpgrid  pub  scripts 	<p>Propriétaire : www-data Sécurité : Lecture</p> <p>Propriétaire : www-data Sécurité : Lecture + Ecriture</p> <p>Propriétaire : www-data Sécurité : Lecture</p> <p>Propriétaire : www-data Sécurité : Lecture + Ecriture</p> <p>Propriétaire : www-data Sécurité : Lecture</p>
---	---

Figure 3: Répertoires du site Web (sécurité)

Modules complémentaires

PHPGrid

PHPGrid est un module commercial qui permet l'élaboration et l'affichage de tableaux (ou listes), utilisant une base de données en tant que source. (Réf : phpgrid.com pour plus d'information).

Il est basé sur PHP, ainsi que le module JavaScript "jqGrid". Ce dernier constitue une extension de la librairie JavaScript "jQuery", qui complète la solution.

orderNumber	orderDate	requiredDate	shippedDate	status	comments	customerNumber	contractDateTime	logTime
10100	2003-01-06	2003-11-13	2003-01-10	Shipped	foo	363		01:10:00
10101	2003-01-18	2003-01-31	2003-01-22	Shipped	Check on availability.	128		12:34:00
10102	2003-01-10	1918-11-23	1946-04-10	Shipped	The cédille (cedilla) Ç ... The Accent aigu (acute accent) é ... The Accent circonflexe (circumflex) â, ê, î, ô, û ... The accent grave (grave accent) à, è, ù ... The accent tréma (dieresis/umlaut) ê, î, ö	181		10:11:02
10103	2003-01-29	2003-02-19	2003-02-02	Shipped	fásdfasdfá asdfá	141		
10104	2003-01-25	2003-02-09	2003-04-24	Shipped		141		
10105	2003-02-11	2003-02-21	2003-02-12	Shipped		145		
10106	2003-02-17	2003-02-24	2003-02-21	Shipped	Fooobar using XML as datasource	278		
10107	2003-02-24	2003-03-03	2003-02-26	Shipped	Difficult to negotiate with customer. We need more marketin	131		00:12:12
10108	2003-03-03	2003-03-12	2008-03-12	Shipped	test	385		16:04:13

Figure 4: Exemple de tableau PHPGrid

phpGridHandle

Dans le cas du site "is.anonyme", une "couche logicielle" a été ajoutée au module "PHPGrid", au travers du développement d'une classe PHP dédiée "phpGridHandle", qui permet la gestion avancée des tableaux gérés par "PHPGrid". Cette classe s'inscrit en concomitance avec MySQL également.

Le principe consiste à stocker les paramètres des divers tableaux dans une table MySQL dédiée et nommée "_grids_definers" (cf: [Table " grids definers"](#)), de façon à pouvoir récupérer ceux-ci relativement facilement et de les appliquer selon les divers cas de figure, par le biais de fonctions PHP, présentes dans la classe.

Elle permet également l'exécution de fonctions particulières selon le type de tableau à représenter.

Exemple de paramètres stockés :

- Les noms des champs d'une table que "PHPGrid" va devoir prendre en charge.
- Le type des champs d'une table (Nombre, Texte, etc.)
- Quels champs doivent être affichés dans le tableau HTML final
- Quels champs traiter lors de l'adjonction d'un nouvel enregistrement dans la table traitée.

Anonyme SA

- L'ordre dans lequel doivent être affichés les champs dans le formulaire d'ajout d'enregistrement.
- Quelle fonction particulière appliquer à tel ou tel tableau, selon les besoins.

Conjointement, une gestion de droits d'accès aux divers tableaux traités par "PHPGrid" a été également développée.

Ces droits d'accès sont issus de Microsoft Active Directory, par l'entremise de groupes de sécurité, créés à cet effet.

LDAP

Pour récupérer les données issues de "Microsoft Active Directory", un module PHP désigné "LDAP" (Lightweight Directory Access Protocol) a été développé en interne et se base sur l'extension du même nom, proposée par PHP.

Le but consiste à stocker certaines informations relatives aux utilisateurs et à certains groupes de sécurité dédiés, issus de l'Active Directory de "Anonyme SA".

Sur le serveur "Ubuntu", il existe deux possibilités d'exécution du module "LDAP" :

1. Via la console du serveur (en ligne de commande)
2. Via le site Web

Mode console

Le mode console d'exécution LDAP est principalement utilisé via l'exécution de tâches automatisées, lancées à intervalles réguliers par le serveur Linux, au travers du module d'exécution de tâches "Crontab".

Il permet :

- De récupérer les modifications intervenues dans l'Active Directory de "Anonyme SA" et de répercuter ces modifications sur le site "is.anonyme"
- Également, il permet d'envoyer des courriels individuellement aux utilisateurs, relativement à l'expiration des mots de passes des comptes Active Directory, par exemple.

En effet, certains utilisateurs ne sont pas authentifiés via un poste de travail standard, lié à l'AD "Anonyme SA". Dans ce cas de figure, un avertissement par le moyen mentionné permet aux utilisateurs d'être prévenus à l'avance de l'expiration de leur mot de passe, afin que ceux-ci entreprennent les mesures nécessaires de changement.

Mode "Web"

Le mode "Web" "LDAP" permet aux administrateurs de la plateforme de consolider la base des utilisateurs du site web, en récupérant manuellement les changements intervenus dans l'Active Directory "Anonyme SA", sans devoir attendre l'exécution horaire du module LDAP.

Cette fonction est exécutée directement via une entrée du menu du site web dans sa version "administration". Elle est protégée par un mot de passe particulier que l'administrateur doit fournir, qui protège l'accès au répertoire où se situe le script PHP afférant.

Base de données

Une base de données a été créée pour servir le site Web "is.anonyme".

Nom de la base : "newisanonyme"

Type de base : MySQL InnoDB (Base relationnelle)

Format de la base : utf8mb4

Les tables

Préfixes de tables

Préfixe	Commentaire
_	Tables contenant des paramètres de gestion du site
_acr	Tables déterminant les droits d'accès au site (A ccess C ontrol R ights), ainsi qu'à d'autres tables.
_ad	Tables en charge de contenir des données issues d' A ctive D irectory
d_	Tables contenant les d onnées à afficher sur le site
s_	Tables utilisées en tant que s ources pour la création de formulaires

Liste des tables

Nom	Usage
_acr_grids	Droits d'accès aux tables en fonction de groupes de sécurité issus d'Active Directory
_acr_grids_fields	Droits d'accès à des champs de tables en fonction de groupes de sécurité issus d'Active Directory (en développement)
_acr_menu	Droits d'accès aux entrées du menu principal du site en fonction de groupes de sécurité issus d'Active Directory
_ad_groups	Groupes de sécurité dédiés issus d'Active Directory
_ad_users	Utilisateurs issus d'Active Directory
_ad_users_groups	Mise en corrélation des utilisateurs et des groupes de sécurité dont ils font partie
_grids_definers	Paramètres stockés en vue de l'affichage de tableaux à l'aide du module "PHPGrid"
_logs	Table de journalisation d'évènements liés au site Web
_menu	Stocke les données relatives au menu principal du site Web

Nom	Usage
<code>_menu_d_documents</code>	Stocke les données relatives à un menu secondaire attendant à la table "d_documents"
<code>d_assets</code>	Table de données relatives à l'inventaire matériel informatique de "Anonyme SA"
<code>d_bdi</code>	Table de données relative à une bibliothèque virtuelle d'ouvrages de Anonyme SA
<code>d_documents</code>	Table de liens relative à des documents liés au Service informatique de "Anonyme SA"
<code>d_hr</code>	En développement
<code>d_hr_totals</code>	En développement
<code>d_loans</code>	Table relative à un système de prêle de matériel informatique
<code>d_news</code>	Table relative à un système de nouvelles internes (en développement)
<code>d_ag_tasks</code>	Table relative à des tâches concernant le groupe "Anonymous Group"
<code>d_softwares</code>	Table de données relatives à l'inventaire de logiciels informatiques de "Anonyme SA"
<code>d_supplies</code>	Tables relative aux fournitures gérées par le Service informatique
<code>d_support</code>	Table relative au support informatique de "Anonyme SA"
<code>s_addresses</code>	"Table source" contenant les adresses des sites géographiques de l'Entreprise "Anonyme SA"
<code>s_categories</code>	"Table source" contenant divers types de matériels ou logiciels
<code>s_groups</code>	"Table source" contenant des groupes de travail issu d'Active Directory. (Ne pas confondre avec les groupes de sécurité utilisés sur le site Web)
<code>s_holidays</code>	"Table source" contenant les jours chômés officiels
<code>s_materials</code>	"Table source" contenant divers types de matériels

Liste des vues

Des vues ont également été créées pour simplifier les requêtes SQL exécutées depuis "PHP".

Nom	Usage
<code>_acr_grids_col_groups</code>	Retrouve les groupes de sécurités assignés à la table " <code>_acr_grids</code> " en tant que champs.
<code>_acr_grids_definers</code>	Visualise des champs issus de la table " <code>_grids_definers</code> " en vue de la gestion de cette même table via le module "PHPGrid"
<code>_acr_grids_fields_col_groups</code>	Retrouve les groupes de sécurités assignés à la table " <code>_acr_grids_fields</code> " en tant que champs
<code>_acr_menu_col_groups</code>	Retrouve les groupes de sécurités assignés à la table " <code>_acr_menu</code> " en tant que champs
<code>_acr_menu_definers</code>	Retrouve des champs issus de la table " <code>_grids_definers</code> " en vue de la gestion de la table " <code>_menu</code> " via le module "PHPGrid"

Procédures stockées

Nom	Usage
<code>getMenuEntry</code>	Retrouve une entrée de menu selon son parent et son identifiant
<code>getToDoInfosFromAGTasks</code>	Retrouve une ligne de la table " <code>d_ag_tasks</code> " selon son identifiant
<code>getUserGroups</code>	Retrouve les groupes de sécurité auxquels appartient un utilisateur donné
<code>getUsersFromGroup</code>	Retrouve les utilisateurs appartenant à un groupe de sécurité
<code>getUsersFullName</code>	Retrouve les nom et prénom d'un utilisateur donné.

Développement du code "PHP"

Page "index.php"

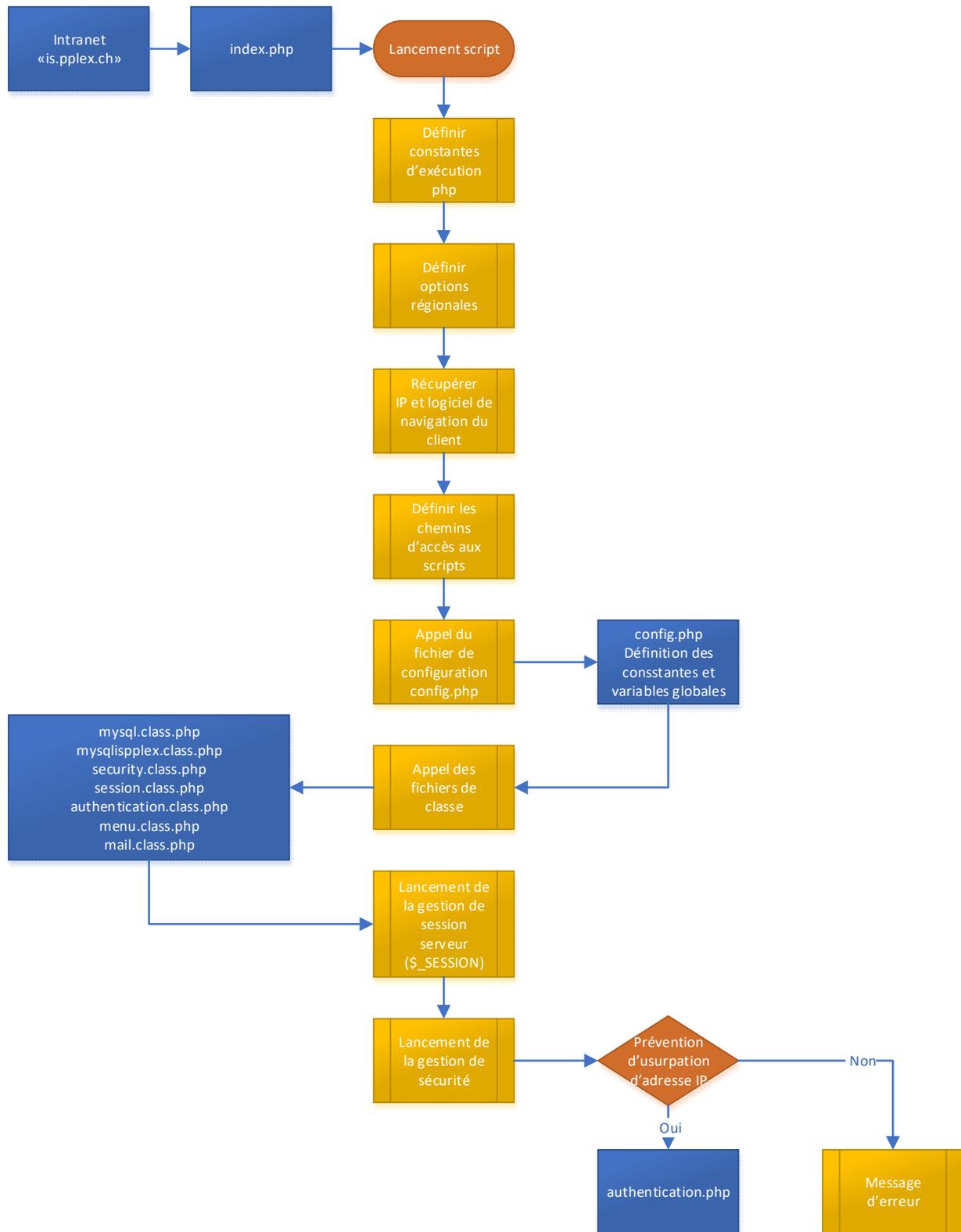


Figure 5: Structure de la page d'index du site Web

Page "authentication.php"

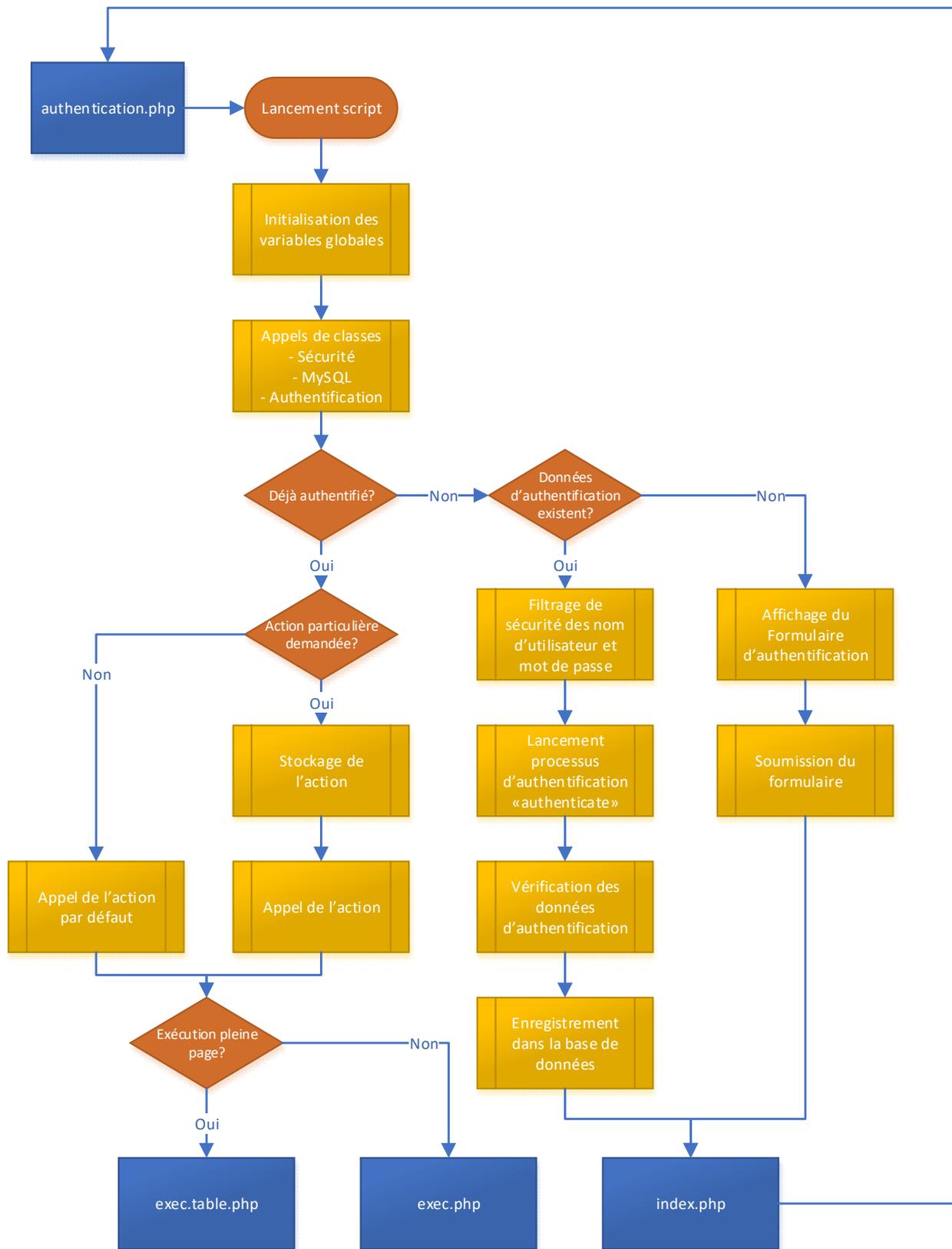


Figure 6: Structure de l'authentification au site Web

Page "exec(.table).php

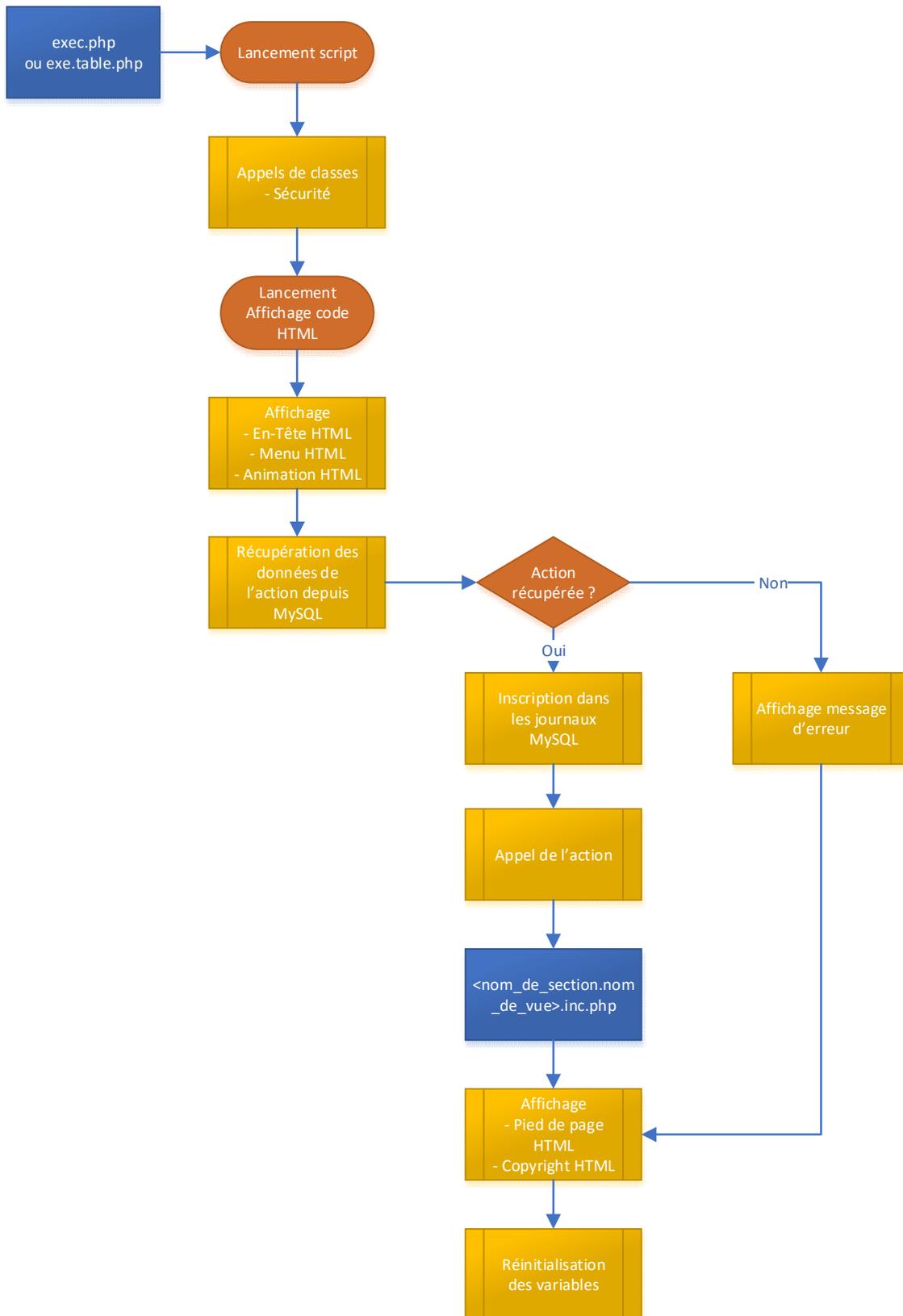


Figure 7: Structure d'exécution d'une section du site Web

Menu principal

Le menu recense la plupart des fonctionnalités proposées par le site "is.anonyme". Il est architecturé selon une table MySQL "_menu".



Figure 8: Représentation du menu

Table "_menu"

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
menuname	VARCHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
position	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
label	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hidden	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
section	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
isparent	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
parent	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
level	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
linkType	SET('hr', 'url', 'uri', 'u...)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'uri'
link	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
gridname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
gridview	SET('full', 'net', 'price'...)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'full'
target	SET('_blank', '_self', '...')	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
align	SET('left', 'right')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'left'
pictures	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
animation	SET('FadeInSS', 'one...')	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figure 9: Descriptif des champs

Description des champs de la table " menu"

Nom	Commentaire
Id	Identifiant unique
menuname	Nom unique du menu
position	Position hiérarchique dans le menu
label	Libellé du menu
hidden	L'entrée de menu est-elle cachée?
section	Numéro de section du site
isparent	L'entrée de menu a-t-elle des un menu enfant (sous-menu)?
parent	Si le menu est un menu enfant, quel est son parent?
level	Niveau de sous-menu (0 menu parent, 1 menu enfant, etc.)
linkType	Type de lien de l'entrée de menu ('hr', 'url', 'uri', 'uriPrint', 'uriTable') <u>Valeurs possibles :</u> hr : Insère un séparateur entre deux entrées de sous-menu url : Lien externe au site ou lien absolu uri : Lien interne au site ou lien relatif uriPrint : Lien interne en vue d'une impression (en développement) uriTable : Lien interne nécessitant une visualisation étendue en termes de largeur de page.
link	Lien interne pointant vers la section du site, ou lien externe
gridname	Nom du tableau "PHPGrid" lié à l'entrée de menu
gridview	Type de visulisation (vue) du tableau "PHPGrid"
target	Cible du lien ('_blank', 'Top_', '_Self')
align	Alignement de l'entrée de menu (left, right)
pictures	Images associées à la section du menu (p.ex. slideshow-01-i.png;slideshow-01-h.png)
animation	Animation associée aux images de la section du site <u>Valeurs possibles :</u> FadeInSS : Apparition graduée des images (en boucle)

Nom	Commentaire
	oneFadeSS : Apparition graduée d'images (non bouclé) riseAnimSS : Apparition d'images, en glissement depuis le bas de page fallAnimSS : Apparition d'images, en glissement depuis le haut de page slideRightSS : Apparition d'images, en glissement depuis la droite noAnimSS : Pas d'animation

Lien interne de type "uri"

Lorsqu'une entrée de menu est un lien typé "uri", cela désigne un renvoi vers l'une des sections du site.

Dans ce cas, le champ "link" est composé comme suit :

<nom_de_section>.<nom_de_visualisation>.inc.php (Exemple : assets.standard.inc.php)

Le lien est ensuite traité par le script PHP "exec.php" ou "exec.table.php", qui recompose celui-ci en adjoignant :

1. L'URL de base du site suivie du chemin "/inc/sections/"
2. Le contenu du champ "link"
3. L'extension ".inc.php"

Classes PHP

Outre les modules externes décrits plus haut, plusieurs classes PHP ont été développées, dédiées à la gestion du site Web.

Fichier de classe	Nom de classe	Commentaires
authentication.class.php	AuthenticationManager	Gère l'authentification au site Web
html.class.php	HTMLCode	Outils de génération et d'affichage de code HTML
mail.class.php	MailManager	Prend en charge la génération et l'envoi de courriels
menu.class.php	MenuManager	Prend en charge l'affichage des menus du site Web
mysql.class.php	sqlConnHandle	Gère l'accès à la base de données MySQL
mysqlisanonyme.class.php	sqlSanonyme	Extension de la classe "sqlConnHandle". Prend en charge l'exécution de fonctions particulières pour la récupération d'informations issues de la base de données et liées au site Web
phpgrid.class.php	phpGridHandle	Classe constituant une couche logicielle supplémentaire et liée au module "PHPGrid", par l'adjonction de fonctions particulières relatives au traitement et à l'affichage de tableaux
security.class.php	checkSecurity	Prend en charge diverses fonctions liées à la sécurité du site Web
session.class.php	SessionManager	Prend en charge la gestion des sessions (cookies) PHP, liés à chaque utilisateur accédant au site Web

Classe "authentication.class.php"

Cette classe regroupe un ensemble de fonctions dédiées à la gestion de l'authentification des utilisateurs. Soit :

1. Authentification d'un utilisateur sur le site Web
2. L'utilisateur est-il authentifié
3. Gestion du mot de passe d'un utilisateur
4. Déconnexion d'un utilisateur

Méthodes de classe

```
I. public function authenticate ($usr, $pwd, $db_hdl, $uri = "")
```

Réalise l'authentification d'un utilisateur sur le site Web.

Variables

\$usr	Nom d'utilisateur	Chaîne de caractères
\$pwd	Mot de passe de l'utilisateur	Chaîne de caractères
\$db_hdl	Variable de gestion de classe "sqlISanonyme"	Non typé
\$uri	Facultatif Selon les cas, stocke le lien demandé par un utilisateur encore non authentifié sur le site. Valeur Par défaut Chaîne vide	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
II. public function isAuthenticated ($db_hdl)
```

Contrôle si l'utilisateur est authentifié sur le site Web.

Variables

\$db_hdl	Variable de gestion de classe "sqlISanonyme"	Non typé
-----------------	--	----------

Valeur retournée par la fonction

- Retourne "Vrai" si l'utilisateur est authentifié ou "Faux" dans le cas contraire (booléen).

Anonyme SA

```
III. public function isPasswordTemporary($db_hdl)
```

Vérifie si le mot de passe de l'utilisateur est temporaire.

Variables

\$db_hdl	Variable de gestion de classe "sqlISanonyme"	Non typé
-----------------	--	----------

Valeur retournée par la fonction

- Retourne "Vrai" si le mot de passe est temporaire ou "Faux" dans le cas contraire (booléen).

```
IV. public function checkPassword($db_hdl, $usr, $pwd)
```

Contrôle si le mot de passe d'un utilisateur est correct et retourne le résultat.

Variables

\$db_hdl	Variable de gestion de classe "sqlISanonyme"	Non typé
\$usr	Nom d'utilisateur	Chaîne de caractères
\$pwd	Mot de passe de l'utilisateur	Chaîne de caractères

Valeur retournée par la fonction

- Retourne "Vrai" si le mot de passe est correct ou "Faux" dans le cas contraire (booléen).

```
V. public function showFormAuthenticate ($uri = "")
```

Affiche le formulaire de connexion au site Web.

Variables

\$uri	Facultatif Selon les cas, stocke le lien demandé par un utilisateur encore non authentifié sur le site. Valeur Par défaut Chaîne vide	Chaîne de caractères
--------------	---	----------------------

Valeur retournée par la fonction

- Aucune

```
VI. public function showFormPwdGen ($db_hdl, $html_hdl, $uri = "")
```

Affiche un formulaire permettant à un administrateur du site la génération déterminée ou aléatoire d'un mot de passe provisoire.

Variables

\$db_hdl	Variable de gestion de classe "sqlISanonyme"	Non typé
\$html_hdl	Variable de gestion de classe "HTMLCode"	Non typé
\$uri	Facultatif Selon les cas, stocke le lien demandé par un utilisateur encore non authentifié sur le site. Valeur Par défaut Chaîne vide	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
VII. public function showFormInfo ($aUser, $encCode, $url, $msg = "", $clr = "")
```

Lorsque l'utilisateur est connecté, affiche les informations relatives à son compte et offre la possibilité de changer le mot de passe en vigueur.

Variables

\$aUser	Données de l'utilisateur courant	Tableau
\$encCode	Chaîne d'encodage issue d'une constante dans le fichier de configuration	Chaîne de caractères
\$url	Url du site Web	Chaîne de caractères
\$msg	Facultatif Message à afficher Valeur Par défaut Chaîne vide	Chaîne de caractères
\$clr	Facultatif Couleur de la police du message au format HTML "#xxxxxx" où x est un chiffre hexadécimal ("0" à "9" ou "a" à "f") Valeur Par défaut Chaîne vide	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
I. private function writeToLogTableFromAuth($db_hdl, $table, $type = "information",
    $op = "", $side = "", $service = "", $action = "", $log = "")
```

Inscrit un évènement dans une de table journalisation MySQL dédiée "_logs", lors d'opérations d'authentification.

Variables

\$db_hdl	Variable de gestion de la classe "sqlISanonyme"	Non typé
\$table	Nom de la table de journalisation	Chaîne de caractères
\$type	Facultatif Type d'évènement, selon un degré d'importance prédéterminé, selon ce qui suit. <u>Valeurs possibles :</u> Par défaut information information Détermine une information warning Requiert de l'attention error Détermine une erreur critical Détermine une erreur critique	Chaîne de caractères
\$op	Facultatif Opération effectuée (p. ex. "CHECKPWD") Valeur Par défaut Chaîne vide	Chaîne de caractères
\$side	Facultatif Si l'action est engendrée par le serveur ou par le client ('Server', 'Client') <u>Valeurs possibles :</u> Par défaut Chaîne vide Server Action initiée par le serveur Client Action initiée par le client	Chaîne de caractères
\$service	Facultatif Service concerné <u>Valeurs possibles :</u> Par défaut Chaîne vide Apache Service Apache LDAP Service LDAP MySQL Service MySQL	Chaîne de caractères

Anonyme SA

	CRONTAB Service CRONTAB POSTFIX Service POSTFIX BROWSING Service de navigation AUTH Service d'authentification SELF Service interne	
\$action	Action effectuée	Chaîne de caractères
\$log	Détail de l'action effectuée	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

Classe "html.class.php"

Cette classe regroupe un ensemble de fonctions dédiées à l'affichage, ainsi qu'à la gestion de code HTML. Soit :

1. Affichage de balises en vue de l'encapsulation de code HTML
2. Préparation et affichage "d'objets" particuliers, tels que des listes d'options HTML, p.ex.
3. Fonctions de transposition de chaînes de caractères

Propriétés de classe

- \$binaryParams

Type:	Tableau
Portée:	Publique
Description:	Propriété regroupant les caractéristiques d'un fichier image, en vue de l'envoi de ce dernier en incrustation binaire dans un courriel.
Contenu par défaut:	<pre>array ('path' => "", 'mime' => "", 'extension' => "")</pre>

Méthodes de classe

```
I. public function showOpenHTMLEnv($site, $dirCSS = "", $dirPIC = "", $dirSCR = "")
```

Affiche l'environnement, ainsi que les "Headers" HTML des pages Web du site.

Variables

\$site	Adresse du site Web	Chaîne de caractères
\$dirCSS	Facultatif Répertoire des feuilles de style du site Web	Chaîne de caractères
\$dirPIC	Facultatif Répertoire des images du site Web Valeur par défaut Chaîne vide	Chaîne de caractères
\$dirSCR	Facultatif Répertoire des scripts du site Web Valeur par défaut Chaîne vide	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

-

```
II. public function showTopMargin()
```

Affiche une marge au sommet de la page Web

Valeur retournée par la fonction

- Aucune

```
III. public function showOpenMainBackground()
```

Affiche l'arrière-plan de la page Web

Valeur retournée par la fonction

- Aucune

```
IV. public function showCloseMainBackground()
```

"Ferme" l'arrière-plan de la page Web

Valeur retournée par la fonction

- Aucune

```
V. public function showCloseHTMLEnv
```

Ferme l'environnement HTML

Valeur retournée par la fonction

- Aucune

```
VI. public function generateOptionList($listID, $datalistID, $anArray, $subKeys = "",  
$subKeysSep = "", $sep = "", $js = "", $class = "clDataList")
```

Génère une liste d'options dans un formulaire avec un choix de correspondance.

Variables

\$listID	Identifiant de la liste HTML	Chaîne de caractères
\$datalistID	Identifiant de la liste HTML, source des éléments à choix	Chaîne de caractères
\$anArray	Tableau des éléments source	Tableau
\$subKeys	Facultatif Ligne de texte indiquant les clefs à utiliser dans le tableau \$anArray. Cette ligne est entrecoupée par un séparateur. Ex. "un/deux/trois/quatre" Valeur par défaut Chaîne vide	Chaîne de caractères

<p>\$sep</p>	<p>Facultatif Le séparateur à utiliser pour séparer \$subKeys en éléments de tableau Ex. "un/deux/trois/quatre"</p> <p>Valeur par défaut Chaîne vide</p>	<p>Chaîne de caractères</p>
<p>\$js</p>	<p>Facultatif Pour insérer du code JavaScript optionnel dans l'élément liste du formulaire</p> <p>Valeur par défaut Chaîne vide</p>	<p>Chaîne de caractères</p>
<p>\$Class</p>	<p>Facultatif Nom de la classe HTML donné à l'attribut "class=" de l'élément liste, en vue de traiter le style d'affichage (CSS), p. ex.</p> <p>Ce paramètre est lié aux fichiers de feuilles de style "CSS", présents sur le site dans le dossier /css/</p> <p>Valeur par défaut clDataList</p>	<p>Chaîne de caractères</p>

Anonyme SA

Valeur retournée par la fonction

- Chaîne de caractères contenant le code HTML généré.

```
VII. public function convertToAsciiCodeForHTML($s)
```

Encode en code ASCII "HTML" une chaîne de caractères.

Variable

<code>\$s</code>	Chaîne de caractères à encoder	Chaîne de caractères
------------------	--------------------------------	----------------------

Valeur retournée par la fonction

- Chaîne de caractères contenant le code HTML généré.

```
VIII. public function getPicturesFromAsciiCode($s, $params = "")
```

Renvoie une série d'images en code HTML (``), associées à l'interprétation d'une chaîne de caractères "\$s". Utilise les images retournées par la fonction "getBinaryData".

Le but est de convertir des fichiers images en code binaire de façon à pouvoir les incruster directement dans du code HTML. Cela évite de devoir télécharger des images lors de la lecture d'un courriel.

Variables

<code>\$s</code>	Chaîne de caractères à transformer	Chaîne de caractères
<code>\$params</code>	Facultatif Paramètres pouvant être ajoutés à la balise HTML "". Valeur Par défaut Chaîne vide	Chaîne de caractères

```
I. private function curlRequest($url)
```

Récupère des données sur la base du lien "\$url" depuis une session "cURL" et les retourne. Utilisé dans la fonction "getBinaryData".

Variable

<code>\$url</code>	Liens pour la récupération des données	Chaîne de caractères
--------------------	--	----------------------

Valeur retournée par la fonction

- Données cURL (non typé)

```
II. private function getBinaryData($file, $mime)
```

Renvoie le code HTML agrémenté des données binaires d'un fichier image en utilisant la fonction "curlRequest".

Variables

\$file	Nom du fichier dont les données doivent être récupérées	Chaîne de caractères
\$mime	Mime/Type du fichier dont les données doivent être récupérées.	Chaîne de caractères

Valeur retournée par la fonction

- Chaîne de caractères agrémentée pour partie encodée en base 64 relative aux données "image" récupérées.

Classe "mail.class.php"

Cette classe regroupe un ensemble de fonctions dédiées à l'élaboration et l'envoi de mail depuis le site web "is.anonyme". Soit :

1. Préparation de corps de messages courriels
2. Envoi de courriels
3. Préparation et envois de courriels dédiés à des groupes d'utilisateurs particuliers

Propriétés de classe

- \$db_ihdl

Type:	Non typé
Portée:	Privée
Description:	Gestionnaire de la base de données interne à la classe.
Contenu par défaut:	Null

- \$sectionParams

Type:	Tableau
Portée:	Publique
Description:	Paramètres appliqués lors de la préparation d'un courriel.
Contenu par défaut:	<pre>array ('title' => "", 'subTitle' => "", 'object' => "", 'afterTextWithLink' => Null)</pre>

- \$tableParams

Type:	Tableau
Portée:	Publique
Description:	Paramètres d'un tableau PHPGrid en vue de préparation d'un courriel.
Contenu par défaut:	<pre>array ('table' => "", 'fields' => "", 'viewType' => "standard", 'recordID' => Null, 'onDeleteString' => Null)</pre>

Anonyme SA

- \$mailParams

Type:	Tableau
Portée:	Publique
Description:	Paramètres généraux d'un courriel à envoyer.
Contenu par défaut:	<pre>array ('to' => Null, 'subject' => Null, 'headers' => array(), 'logo' => Null, 'signature' => Null, 'company' => Null, 'dpt' => Null)</pre>

- \$mailPwdExpParams

Type:	Tableau
Portée:	Publique
Description:	Paramètres complémentaires d'un courriel de type "Expiration de mot de passe".
Contenu par défaut:	<pre>array ('daysExpire' => Null, 'dateExpire' => Null, 'link' => Null, 'lang' => Null)</pre>

- \$mailPwdInit

Type:	Tableau
Portée:	Publique
Description:	Paramètres complémentaires d'un courriel de type "Réinitialisation de mot de passe".
Contenu par défaut:	<pre>array ('daysExpire' => Null, 'dateExpire' => Null, 'link' => Null, 'lang' => Null)</pre>

Anonyme SA

- \$mailICalEvent

Type:	Tableau
Portée:	Publique
Description:	Paramètres d'un objet "iCalendar".
Contenu par défaut:	<pre>array ('subject' => Null, 'ToDo' => array(), 'fromUser' => array(), 'toUsers' => array(), 'mailSubject' => Null, 'mailSignature' => Null, 'domain' => Null)</pre>

- \$unwantedChars

Type:	Tableau
Portée:	Privée
Description:	Tableau de caractères à substituer dans le sujet d'un courriel lors de son envoi, du fait d'un encodage de caractères non supportés.
Contenu par défaut:	<pre>array('Š'=>'S', 'š'=>'s', 'Ž'=>'Z', 'ž'=>'z', 'À'=>'A', 'Á'=>'A', 'Â'=>'A', 'Ã'=>'A', 'Ä'=>'A', 'Å'=>'A', 'Æ'=>'A', 'Ç'=>'C', 'È'=>'E', 'É'=>'E', 'Ê'=>'E', 'Ë'=>'E', 'Ì'=>'I', 'Í'=>'I', 'Î'=>'I', 'Ï'=>'I', 'Ñ'=>'N', 'Ò'=>'O', 'Ó'=>'O', 'Ô'=>'O', 'Õ'=>'O', 'Ö'=>'O', 'Ø'=>'O', 'Ù'=>'U', 'Ú'=>'U', 'Û'=>'U', 'Ü'=>'U', 'Ý'=>'Y', 'ß'=>'B', 'ß'=>'Ss', 'à'=>'a', 'á'=>'a', 'â'=>'a', 'ã'=>'a', 'ä'=>'a', 'å'=>'a', 'æ'=>'a', 'ç'=>'c', 'è'=>'e', 'é'=>'e', 'ê'=>'e', 'ë'=>'e', 'ì'=>'i', 'í'=>'i', 'î'=>'i', 'ï'=>'i', 'ö'=>'o', 'ñ'=>'n', 'ò'=>'o', 'ó'=>'o', 'ô'=>'o', 'õ'=>'o', 'ö'=>'o', 'ø'=>'o', 'ù'=>'u', 'ú'=>'u', 'û'=>'u', 'ý'=>'y', 'ß'=>'b', 'ÿ'=>'y')</pre>

- \$dbDatas

Type:	Tableau
Portée:	Privée
Description:	Paramètres d'un objet "iCalendar".
Contenu par défaut:	<pre>Array(array())</pre>

Méthodes de classe

```
I. public function __construct($db_h = Null)
```

Initialise la classe "MailManager" lors de l'appel de la classe. Applique une valeur à la propriété interne "\$db_ihdl".

Variable

<code>\$db_h</code>	Variable de gestion de classe "sqlISanonyme"	Non typé
---------------------	--	----------

Valeur retournée par la fonction

- Aucune

```
II. public function prepareEMailMessage()
```

Prépare le contenu d'un courriel envoyé lors d'opérations effectuées sur des tables de la base de données et prises en charge par le module PHPGrid, telles que l'ajout, la modification ou encore la suppression d'enregistrements d'une table.

Valeur retournée par la fonction

- Chaîne de caractères contenant le corps du message du courriel

```
III. public function prepareEMailMessageForAGGroup()
```

Prépare le contenu d'un courriel envoyé lors d'opérations effectuées sur des tables de la base de données et prises en charge par le module PHPGrid, telles que l'ajout, la modification ou encore la suppression d'enregistrements d'une table. Cette fonction est spécifiquement dédiée à un groupe de travail particulier.

Valeur retournée par la fonction

- Chaîne de caractères contenant le corps du message du courriel

```
IV. public function prepareEMailMessageForPwdExp()
```

Prépare le contenu d'un courriel envoyé lorsqu'un mot de passe doit être modifié dans l'Active Directory "Anonyme SA".

Valeur retournée par la fonction

- Chaîne de caractères contenant le corps du message du courriel

```
V. public function prepareEMailMessageForPwdReset()
```

Prépare le contenu d'un courriel envoyé lors de l'établissement d'un mot de passe provisoire pour une utilisateur de la plateforme "is.anonyme".

Valeur retournée par la fonction

- Chaîne de caractères contenant le corps du message du courriel

Fonctions liées à des groupe particuliers

```
VI. public function getAGTasksForToUsers($db_hdl)
```

Récupère les données d'utilisateurs destinataires, membres du groupe ""Anonymous Group"" en vue leur adresser un courriel.

Variable

<code>\$db_h</code>	Variable de gestion de classe "sqlISanonyme"	Non typé
---------------------	--	----------

Valeur retournée par la fonction

- Aucune

```
VII. public function prepareEMailMessageForIcalEvent($count, $method = 'REQUEST')
```

Prépare le contenu d'un courriel envoyé à un groupe de travail particulier et contenant un objet "iCalendar", destiné à être inscrit dans un calendrier Microsoft Outlook.

Variables

<code>\$count</code>	Compteur définissant le nombre de courriels à préparer en cas de destinataires multiples	Entier numérique
<code>\$method</code>	Facultatif Cette variable définit la méthode de l'objet "iCalendar" associée à l'objet calendrier. <u>Conseil :</u> Ne pas modifier <u>Valeurs possibles :</u> Par défaut REQUEST REQUEST PUBLISH	Chaîne de caractères

Valeur retournée par la fonction

- Chaîne de caractères contenant le corps de l'objet "iCalendar"

Fonctions liées à l'envoi de courriels

```
VIII. public function sendMail($messageType = "", $message = "")
```

Envoie un courriel selon un type spécifié.

Variables

<p>\$messageType</p>	<p>Facultatif Type de courriel à envoyer</p> <p><u>Valeurs possibles :</u> Par défaut Chaîne vide phpGrid Pour traiter l'envoi d'un courriel lors de l'ajout/modification/suppression d'un élément traité par le module "phpGrid" PasswordExpire Pour traiter l'envoi d'un courriel relatif à l'expiration d'un mot de passe PasswordInit Pour traiter l'envoi d'un courriel lors de l'initialisation d'un mot de passe sur le site "is.anonyme" AGGroup Pour traiter l'envoi d'un courriel lors de l'ajout/modification/suppression d'un élément traité par le module "phpGrid" spécifique au "groupe "Anonymous Group""</p>	<p>Entier numérique</p>
<p>\$message</p>	<p>Facultatif Contenu du courriel à envoyer</p> <p>Par défaut Chaîne vide</p>	<p>Chaîne de caractères</p>

Valeur retournée par la fonction

- "Vrai" en cas d'envoi de courriel confirmé, "Faux" en cas d'envoi avorté (booléen).

```
IX. public function sendMailReminders($messageType = "")
```

Envoie un courriel relatif à objet "iCalendar".

Variable

\$messageType	Facultatif Type de courriel à envoyer <u>Valeurs disponibles :</u> Par défaut Chaîne vide AGGroupReminder Envoi d'objets "iCalendar" aux utilisateurs membres du groupe ""Anonymous Group""	Chaîne de caractères
----------------------	--	----------------------

Valeur retournée par la fonction

- "Vrai" en cas d'envoi de courriel confirmé, "Faux" en cas d'envoi avorté (booléen).

Fonctions utilitaires

```
X. public function getMailAddress($user)
```

Renvoie l'adresse de courriel d'un utilisateur depuis la base de données

Variable

<code>\$user</code>	Nom d'utilisateur	Chaîne de caractères
---------------------	-------------------	----------------------

Valeur retournée par la fonction

- Chaîne de caractères contenant une adresse de courriel

```
I. private function getTableDatas()
```

Retrouve, sous forme de tableau, les noms de champs et de table référencés dans la table MySQL "_grids_definers", qui englobe les différents paramètres nécessaires à appliquer au module "PHPGrid".

Une fois les noms de champ récupérés : Permet dans un second temps de retrouver des données issues d'autres tables MySQL, contenant des données spécifiques à insérer dans le corps d'un courriel en préparation.

Valeur retournée par la fonction

- Aucune
- Les valeurs obtenues sont attribuées directement à la propriété de classe "\$dbDatas".

Classe "menu.class.php"

Cette classe regroupe un ensemble de fonctions dédiées la gestion du menu principal du site Web "is.anonyme". Soit :

1. Affichage d'éléments de menu
2. Attribution de liens aux entrées de menu

Méthodes de classe

```
I. public function showBeginTopMenuSection($section, $rowAlign, $rowSection)
```

Affiche le début de section du menu et détermine quelle section du site est active pour mettre en surbrillance l'entrée de menu concernée.



Figure 10: Mise en surbrillance du menu en regard de la page courante, affichée

Variables

\$section	La section (page) en cours du site	Chaîne de caractères
\$rowAlign	Selon l'entrée de menu, le paramètre d'alignement "CSS" de cette même entrée	Chaîne de caractères
\$rowSection	La section (page) du site	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
II. public function showEndTopMenuSection()
```

Affiche la fin de la section du menu

Valeur retournée par la fonction

- Aucune

```
III. public function showBeginSubMenuSection()
```

Affiche le début de section d'un sous-menu

Valeur retournée par la fonction

- Aucune

```
IV. public function showEndSubMenuSection()
```

Affiche la fin de la section d'un sous-menu

Valeur retournée par la fonction

- Aucune

```
V. public function showTopMenuItem()
```

Affiche le menu déroulant et met en surbrillance l'élément de menu survolé par la souris.

Valeur retournée par la fonction

- Aucune



Figure 11: Élément du menu principal mis en surbrillance

```
VI. public function showSubMenuItem()
```

Mise en surbrillance d'un élément de sous-menu

Valeur retournée par la fonction

- Aucune

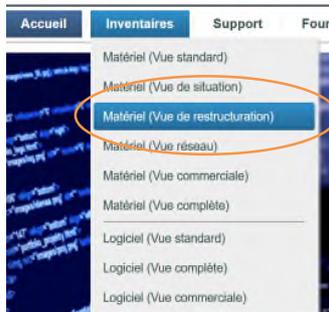


Figure 12: Élément de sous-menu mis en surbrillance

```
VII. public function showMenuInter($count, $result, $align = "")
```

Affiche un séparateur vertical entre deux éléments du menu principal



Variables

\$count	Numéro de l'entrée de menu	Nombre entier
\$result	Variable contenant les entrées issues de la table MySQL "_menu"	Tableau
\$align	Variable d'alignement facultative Valeurs possibles : Par défaut Chaîne vide left Alignement à gauche right Alignement à droite	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
I. private function setLink($row)
```

Vérifie si le lien attaché à une entrée de menu est une URL ou non. S'il s'agit d'une URL, le lien est retourné tel quel, sinon, il est recomposé de façon à constituer une URL complète.

Variable

\$row	Ligne complète de la table MySQL "_menu"	Tableau
--------------	--	---------

Valeur retournée par la fonction

- Chaîne de caractères composant une URL.

Classe "mysql.class.php"

Cette classe regroupe un ensemble de fonctions dédiées à l'accès au serveur MySQL. Soit :

1. Authentification au serveur MySQL
2. Fonctions dédiées à la gestion des tables MySQL

Propriétés de classe

- `$_SQLHandle = Null;`

Type:	Non typé
Portée:	Publique
Description:	Gestion d'accès à MySQL
Contenu par défaut:	Null

Méthodes de classe

```
I. public function __construct($_MYSQL)
```

Initialise la classe en réalisant la connexion au serveur MySQL.

Variable

\$_MYSQL	Paramètres de connexion au serveur MySQL	Tableau
-----------------	--	---------

Valeur retournée par la fonction

- Aucune

```
II. public function getPfxTables($tablePrefix)
```

Retourne une liste des tables de la base de données, selon le préfixe de la table

Variable

\$tablePrefix	Préfix de table MySQL	Chaîne de caractères
----------------------	-----------------------	----------------------

Valeur retournée par la fonction

- Retourne un tableau contenant les tables préfixées par "\$tablePrefix"

```
III. public function query($query)
```

Réécriture de la fonction "mysqli_query". Lance la requête selon la chaîne "\$query" ou stoppe le déroulement script PHP si une erreur survient.

Variable

\$query	Requête MySQL	Chaîne de caractères
----------------	---------------	----------------------

Valeur retournée par la fonction

- Retourne le gestionnaire issu de la requête.

```
IV. public function fetch_assoc($query)
```

Réécriture de la fonction "mysqli_fetch_assoc".

Variable

\$query	Requête MySQL	Chaîne de caractères
----------------	---------------	----------------------

Valeur retournée par la fonction

- Retourne un lot résultat sous forme de tableau associatif, selon la requête MySQL "\$query".

```
V. public function saveTableData($tableName, $aData = array(), $condition = "")
```

Effectue une requête MySQL selon le nom d'une table, des champs de table et des conditions SQL (p. ex. [...]WHERE \$id=[...])

Variables

\$tableName	Nom de la table	Chaîne de caractères
\$aData	Facultatif Tableau de clefs Par défaut Tableau vide	Tableau
\$condition	Facultatif Variable permettant l'application de conditions adjointes à la requête MySQL Par défaut Chaîne vide	Chaîne de caractères

Valeur retournée par la fonction

- Tableau associatif de valeurs dont les clefs sont contenues dans "\$aData".

```
VI. public function resetTable($tableName)
```

Vide le contenu d'une table MySQL

Variable

\$tableName	Nom de la table à vider	Chaîne de caractères
--------------------	-------------------------	----------------------

Valeur retournée par la fonction

- Aucune

```
VII. public function resetFKTable($tableName)
```

Supprime les enregistrements d'une table MySQL et réinitialise le compteur de ligne à 0.

Variable

<code>\$tableName</code>	Nom de la table à vider	Chaîne de caractères
--------------------------	-------------------------	----------------------

Valeur retournée par la fonction

- Aucune

```
VIII. public function modSafeUpdateMode($mode)
```

Active ou désactive le mode sécurisé. Ce mode permet, si besoin, des opérations à grande échelle sur les tables (ex. supprimer l'ensemble des enregistrements d'une table).

Par défaut MySQL interdit des opérations à large échelle, par mesure de sécurité.

Le mode sécurisé doit être désactivé avant de lancer la requête, puis réactivé après.

Variable

<code>\$mode</code>	Active ou désactive le mode sécurisé MySQL. Valeurs possibles : True Active le mode sécurisé False Désactive le mode sécurisé	Booléen
---------------------	--	---------

Valeur retournée par la fonction

- Aucune

```
IX. public function sql_disconnect()
```

Déconnexion du serveur MySQL.

Valeur retournée par la fonction

- Aucune

```
I. private function show_errors($err)
```

Affiche une erreur à l'écran.

Variable

<code>\$err</code>	Erreur à afficher	Chaîne de caractères
--------------------	-------------------	----------------------

Valeur retournée par la fonction

- Aucune

Classe "mysqlisanonyme.class.php"

Cette classe constitue une extension de la classe "mysql.class.php" et regroupe un ensemble de fonctions orientées MySQL, dédiées au site "is.anonyme".

Méthodes de classe

```
I. public function writeToLogTable ($table, $type = "information", $op = "", $side = "", $service = "", $action = "", $log = "")
```

Inscrit des événements dans la table "_logs" de journalisation MySQL de la base de données "newisanonyme".

Variables

\$table	Nom de la table de journalisation	Chaîne de caractères
\$type	Facultatif Type d'évènement, selon un degré d'importance prédéterminé, selon ce qui suit. <u>Valeurs possibles :</u> Par défaut information information Détermine une information warning Requiert de l'attention error Détermine une erreur critical Détermine une erreur critique	Chaîne de caractères
\$op	Facultatif Opération effectuée (p. ex. "CHECKPWD") Valeur Par défaut Chaîne vide	Chaîne de caractères
\$side	Facultatif Si l'action est engendrée par le serveur ou par le client ('Server', 'Client') <u>Valeurs possibles :</u> Par défaut Chaîne vide Server Action initiée par le serveur Client Action initiée par le client	Chaîne de caractères
\$service	Facultatif Service concerné <u>Valeurs possibles :</u> Par défaut Chaîne vide Apache	Chaîne de caractères

Anonyme SA

	Service Apache LDAP Service LDAP MySQL Service MySQL CRONTAB Service CRONTAB POSTFIX Service POSTFIX BROWSING Service de navigation AUTH Service d'authentification SELF Service interne	
\$action	Action effectuée	Chaîne de caractères
\$log	Détail de l'action effectuée	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
II. public function addDBColumnToTableVsLDAP($table, $col, $type, $default = "NULL")
```

Ajout d'une colonne "\$col" dans la table "\$table" de type "\$type" avec une valeur par défaut optionnelle "\$default". Utilisé dans le contexte de mise à jour de tables MySQL via des requêtes "LDAP".

Variables

\$table	Nom de la table	Chaîne de caractères
\$col	Nom de la colonne	Chaîne de caractères
\$type	Type de la colonne	Chaîne de caractères
\$default	Valeur par défaut de la colonne Par défaut NULL	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
III. public function deleteDBColumnFromTableVsLDAP($table, $col)
```

Suppression d'une colonne "\$col" dans la table "\$table". Utilisé dans le contexte de mise à jour de tables MySQL via des requêtes "LDAP".

Variables

\$table	Nom de la table	Chaîne de caractères
\$col	Nom de la colonne	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
IV. public function getDataFrom_grids_definersTableView($table, $field = '')
```

Effectue une requête sur une table MySQL selon un champ "\$field". Si la valeur de "\$field" est vide, retourne une ligne entière de la table "\$table". Utilisé dans le contexte "PHPGrids".

Variables

\$table	Nom de la table	Chaîne de caractères
\$field	Facultatif Nom du champ Par défaut Chaîne vide	Chaîne de caractères

Valeur retournée par la fonction

- Tableau

```
V. public function colNameExistsIn_grids_definers($bench, $field)
```

Recherche une ligne de la table "_ grids_definers", selon "\$bench" et "\$field". Utilisé dans le contexte d'utilisation du module "PHPGrids".

Variables

\$bench	Nom d'une table MySQL	Chaîne de caractères
\$field	Nom du champ	Chaîne de caractères

Valeur retournée par la fonction

- Une ligne de la table "_grid_definers" ou "**false**" si aucune ligne n'est trouvée.

```
VI. public function addColNameInRowTo_grids_definers($bench,$colName)
```

Ajoute une ligne à la table "`_ grids_definers`" en fin de table, en spécifiant des valeurs par défaut aux champs de la table "`_ grids_definers`", en regard de la ligne ajoutée. Utilisé dans le contexte d'utilisation du module "PHPGrids".

Variables

\$bench	Nom d'une table	Chaîne de caractères
\$field	Nom de champ de la table "\$bench"	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
VII. public function deleteColAsRowFrom_grids_definers($row)
```

Suppression d'une ligne de la table "`_ grids_definers`", selon un tableau associatif contenant des paramètres tels que les noms d'une table et d'une colonne. Utilisé dans le contexte d'utilisation du module "PHPGrids".

Variable

\$row	Tableau associatif	Tableau
--------------	--------------------	---------

Valeur retournée par la fonction

- Aucune

```
VIII. public function getDataFrom_acr_TableView($table)
```

Effectue une requête sur une table et récupère les noms des colonnes de la table. Utilisé dans le contexte de gestion de sécurité du site Web.

Variable

\$table	Nom d'une table	Chaîne de caractères
----------------	-----------------	----------------------

Valeur retournée par la fonction

- Tableau contenant le nom des colonnes de la table.

IX. public function getUserGroupsACR(\$usr)

Fait appel à une procédure stockée MySQL (getUserGroups), selon le nom d'utilisateur "\$usr". Utilisé dans le contexte de gestion de sécurité du site Web.

Variable

\$usr	Nom d'utilisateur	Chaîne de caractères
--------------	-------------------	----------------------

Valeur retournée par la fonction

- Tableau contenant le nom des groupes de sécurité dont l'utilisateur est membre.

X. public function getEffectiveACRMenu(\$usr)

Recherche les entrées du menu principal auxquels l'utilisateur connecté à accès. Utilisé dans le contexte de gestion de sécurité du site Web.

Variable

\$usr	Nom d'utilisateur	Chaîne de caractères
--------------	-------------------	----------------------

Valeur retournée par la fonction

- Tableau contenant le nom des menus auxquels l'utilisateur a accès.

XI. public function setGridACR(\$usr, \$bench)

Applique les droits d'accès à une table donnée (\$bench), selon l'utilisateur courant. Utilisé dans le contexte de gestion de sécurité du site Web, ainsi que de la gestion "PHPGrid".

Variables

\$usr	Nom d'utilisateur	Chaîne de caractères
\$bench	Nom d'une table de la base de données	Chaîne de caractères

Valeur retournée par la fonction

- Retourne une chaîne de caractères spécifiant les droits d'opérer sur une table de la base de données, tels que:
 - o Création (**C**reate)
 - o Lecture (**R**ead)
 - o Modification (**U**ppdate)
 - o Suppression (**D**eleate)

La fonction retourne p. ex. "CRUD" (**C**reate, **R**ead, **U**ppdate, **D**eleate)

```
XII. public function getUsersFromDB ()
```

Récupère les utilisateurs de la base de données et les retourne sous forme de tableau.

Valeur retournée par la fonction

- Retourne un tableau associatif contenant des informations relatives aux utilisateurs inscrits dans la table "_ad_users". On obtient ainsi les nom, prénom, adresse de courriel, nom d'utilisateur.

```
XIII. public function getUserFromDB ($usr)
```

Recherche des informations sur un utilisateur donné.

Variable

\$usr	Nom d'utilisateur	Chaîne de caractères
--------------	-------------------	----------------------

Valeur retournée par la fonction

- Retourne un tableau associatif contenant des informations relatives à un utilisateur donné, inscrit dans la table "_ad_users". On obtient ainsi les nom, prénom, adresse de courriel, nom d'utilisateur.

```
XIV. public function setNewUserPasswordInDB ($usr, $pwd, $isTemp, $cost=8)
```

Inscrit un mot de passe dans la table des utilisateurs, pour un utilisateur donné. Utilisé dans le contexte de gestion de sécurité du site Web

Variables

\$usr	Nom d'utilisateur	Chaîne de caractères
\$pwd	Mot de passe	Chaîne de caractères
\$isTemp	Si le mot de passe est temporaire	Booléen
\$cost	Facultatif Coût algorithmique qui doit être utilisé par la fonction PHP "password_hash". Détermine les conditions de hachage d'un mot de passe selon les capacités de calcul du serveur hôte. <u>Valeurs possibles :</u> Par défaut 8 Entier numérique calculé sur la base des capacités de calcul du serveur. (Cf. fonction "password_hash" de la documentation "PHP")	Entier numérique

Valeur retournée par la fonction

- Aucune

```
XV. public function verifyNewUserPasswordInDB($usr, $pwd)
```

Vérifie le nouveau mot de passe d'un utilisateur. Utilisé dans le contexte de gestion de sécurité du site Web.

Variables

\$usr	Nom d'utilisateur	Chaîne de caractères
\$pwd	Mot de passe	Chaîne de caractères

Valeur retournée par la fonction

- Retourne "**true**" si le mot de passe est correct ou "**false**" si le mot de passe ne correspond pas.

```
XVI. public function getTableValueVsID($table, $field, $id)
```

Récupère la valeur d'une table "\$table", en fonction du champ "\$field" et de l'identifiant de la ligne de la table "\$id". Utilisé dans le contexte de l'utilisation du module "PHPGrid".

Variables

\$table	Nom d'utilisateur	Chaîne de caractères
\$field	Mot de passe	Chaîne de caractères
\$id	Numéro de ligne dans la table	Entier numérique

Valeur retournée par la fonction

- Retourne la valeur du champ spécifié de la table ou "**false**" si la requête échoue.

```
XVII. public function getTableValueVsFieldValue($table, $getFields, $field, $fieldVal,
    $isStr = false)
```

Récupère les lignes d'une table selon la clause MySQL "WHERE". Utilisé dans le contexte de l'utilisation du module "PHPGrid".

Variables

\$table	Nom d'une table	Chaîne de caractères
\$getFields	Champs à récupérer de la table	Chaîne de caractères
\$field	Champ ajouté à la clause "WHERE"	Chaîne de caractères
\$fieldVal	Valeur assignée à la clause "WHERE"	Chaîne de caractères
\$isStr	Facultatif Spécifie si "\$fieldVal" est un entier numérique <u>Valeurs possibles :</u> Par défaut false false true	Booléen

Valeur retournée par la fonction

- Retourne la valeur du champ spécifié de la table "\$table", ou "**false**" si la requête échoue.

```
XVIII. public function getAllTableValuesVsID($table, $id)
```

Recherche une ligne d'une table MySQL selon son identifiant unique. Utilisé dans le contexte de l'utilisation du module "PHPGrid".

Variables

\$table	Nom d'utilisateur	Chaîne de caractères
\$id	Numéro de ligne dans la table	Entier numérique

Valeur retournée par la fonction

- Retourne une ligne de la table spécifiée, ou false si la requête échoue.

```
XIX. public function setTableValueVsID($table, $field, $id, $val)
```

Modifie la valeur d'un champ d'une table. Utilisé dans le contexte de l'utilisation du module "PHPGrid".

Variables

\$table	Nom d'utilisateur	Chaîne de caractères
\$field	Champ de la table	Chaîne de caractères
\$id	Numéro identifiant d'une ligne dans la table	Entier numérique
\$val	Valeur du champ	Chaîne de caractères

Valeur retournée par la fonction

- Retourne **"true"** si la requête réussit ou **"false"** si la requête échoue.

```
XX. public function getToDoInfosFromAGTasksVsID($userKeys,$idAGT)
```

Exécute une requête via une procédure stockée MySQL "getToDoInfosFromAGTasks". Utilisé dans le contexte de l'utilisation du module "PHPGrid" ("iCalendar"), pour le groupe d'utilisateurs ""Anonymous Group"".

Variables

\$userKeys	Valeurs à employer dans le tableau retourné	Chaîne de caractères
\$idAGT	Identifiant d'une ligne de la table du groupe de travail	Chaîne de caractères

Valeur retournée par la fonction

- Retourne un tableau associatif à deux dimensions, identifiant les utilisateurs qui sont assignés à la tâche retourne par la procédure stockée.

```
XXI. public function getDocumentsTree ()
```

Retourne un tableau relatif à une arborescence utilisée dans la section "Documents" du site Web. (En développement)

```
I. private function getGridACRvsUser($usr,$bench)
```

Retrouve les droits d'accès à une table (Cf. [fonction "setGridACR"](#)). Utilisé dans le contexte de gestion de sécurité du site Web.

Variables

\$usr	Nom de l'utilisateur courant	Chaîne de caractères
\$bench	Nom d'une table	Chaîne de caractères

Valeur retournée par la fonction

Anonyme SA

- Retourne un entier numérique entre 0 et 15. "0" symbolise le droit d'accès le plus faible et "15", le plus élevé.

Classe "phpgrid.class.php"

Cette classe constitue une couche logicielle supplémentaire par rapport au module "PHPGrid" et regroupe un ensemble de fonctions dédiées à systématiser et à particulariser la gestion et l'affichage du module. En particulier, vis-à-vis de l'accès aux tables MySQL dédiées à ce module.

Propriétés de classe

- `private static $_dbh = Null;`

Type:	Non typé
Portée:	Privée
Description:	Variable de gestion MySQL interne à la classe.
Contenu par défaut:	Null

- `private static $_pgh = Null;`

Type:	Non typé
Portée:	Privée
Description:	Variable de gestion "PHPGrid" interne à la classe.
Contenu par défaut:	Null

- `public $_PGEEnvParams`

Type:	Tableau
Portée:	Publique
Description:	Agrège plusieurs paramètres de gestion du module "PHPGrid".
Contenu par défaut:	<pre>array ('urlSite' => Null, 'dirDOCS' => Null, 'userGroupForSelect' => "", 'sqlTable' => Null, 'viewType' => Null, 'queryFilter' => "", 'tableSort' => array ('field' => "", 'direction' => ""), 'tableWidth' => Null, 'tableShowCaption' => true, 'tableSetToolbarOnTop' => false, 'tableExport' => array('format' => Null,</pre>

```
        'props' => Null
      ),
      'tableEnableSearch' => false,
      'tableGroupBy' => Null,
      'tableGroupSummary' => array(
        'field' => Null,
        'op' => Null
      ),
      'tableColsWidths' => array (
        'Cols' => Null,
        'Widths' => Null
      ),

      'formWidth' => Null,
      'aformColRequired' => array (),
      'formColCreateDefValues' => array(),
      'formColEditDefValues' => array(),

      'enableEdit' => array (
        'type' => "",
        'options' => ""
      ),

      'html' => array(),

      'currency' => array(),

      'groupsToCheck' => array(),

      'stringForDelete' => ""
    )
  )
}
```

Méthodes de classe

```
I. public function __construct($sqlTable = Null, $viewType = Null, $db_h = Null, $pg_h = Null)
```

Initie la classe et affecte les variables fournies à plusieurs propriétés internes.

Variables

\$sqlTable	Facultatif Nom de la table MySQL à traiter <u>Valeurs possibles :</u> Par défaut Null Chaîne de caractères	Chaîne de caractères
\$viewType	Facultatif Type de visualisation de la table (Affecte ce qui sera présenté au client) <u>Valeurs possibles :</u> Par défaut Null Chaîne de caractères	Chaîne de caractères
\$db_h	Facultatif Variable de Gestion "MySQL" <u>Valeurs possibles :</u> Par défaut Null Gestionnaire "MySQL"	Non typé
\$pg_h	Facultatif Variable de Gestion "PHPGrid" <u>Valeurs possibles :</u> Par défaut Null Gestionnaire "PHPGrid"	Non typé

Valeur retournée par la fonction

- Aucune

II. public function openHtmlEnvironment(\$supHtml = "")

Construit et affiche le contenu d'une page HTML en ouvrant des balises et en vue l'affichage d'une table MySQL sous forme de tableau.

Intègre des balises de titres HTML (<h1> à <h6>), renseignées au travers de la variable "\$_PGEParams['html']", ainsi que du texte optionnel, puis fixe la largeur d'affichage du tableau "PHPGrid" via une balise "<div>".

A partir du niveau "h5", se comporte comme un sous-tableau, intégré dans une boucle, permettant l'insertion du titre "<h5>", suivi de lignes de texte "standard".

Exemple :

```
$gestionnairePHPGrid -> _PGEParams['html']['h1'] = "Titre 1"
$gestionnairePHPGrid -> _PGEParams['html']['h4'] = "Titre 4"
$gestionnairePHPGrid -> _PGEParams['html']['h5'][0] = "Titre 50"
$gestionnairePHPGrid -> _PGEParams['html']['h5'][0]['html'][0] = "Texte 50.0"
$gestionnairePHPGrid -> _PGEParams['html']['h5'][0]['html'][1] = "Texte 50.1"
$gestionnairePHPGrid -> _PGEParams['html']['h5'][1] = "Titre 51"
$gestionnairePHPGrid -> _PGEParams['html']['h5'][1]['html'][0] = "Texte 51.0"
$gestionnairePHPGrid -> _PGEParams['html']['h5'][1]['html'][1] = "Texte 51.1"
```

Variable

\$supHtml	<p>Variable d'insertion de code HTML ou texte supplémentaire après les balises de titre et avant l'affichage du tableau "PHPGrid" proprement dit.</p> <p><u>Valeurs possibles :</u> Par défaut Chaîne vide</p> <p>Chaîne de caractères</p>	Chaîne de caractères
------------------	--	----------------------

- \$supHtml : Variable texte pour du code HTML éventuel supplémentaire.

Valeur retournée par la fonction

- Aucune

III. public function closeHtmlEnvironment()

Ferme les balises du contenu HTML, ouvertes par la fonction "openHtmlEnvironment".

Valeur retournée par la fonction

- Aucune

IV. public function setHtmlJQGridStyle ()

Affecte des paramètres d'affichage de base prédéfinis via la balise HTML "<style>", aux colonnes d'un tableau "PHPGrid" à afficher.

Valeur retournée par la fonction

- Aucune

V. public function setHtmlDivWidthConstraint ()

Insère du code JavaScript en vue de contraindre la largeur du tableau "PHPGrid" à afficher.

Valeur retournée par la fonction

- Aucune

VI. public function setHtmlToggleToolbar ()

Insère du code JavaScript afin d'afficher des champs de recherche au sommet de chaque colonne d'un tableau "PHPGrid".

Valeur retournée par la fonction

- Aucune

VII. public function setExportOptions ()

Affiche une barre d'outils sous forme d'icônes ou de liste, donnant des possibilités d'export du tableau affiché courant, selon les formats de fichiers PDF, XLS, CSV.

Les paramètres d'export sont fournis par la variable :

```
$gestionnairePHPGrid -> _PGEParams['tableExport']
```

Valeur retournée par la fonction

- Aucune

VIII. public function setGridSizeFromMenuViewStyle ()

Détermine le paramètre de largeur d'affichage d'un tableau, selon que celui-ci nécessite une présentation plus ou moins large (soit en mode de présentation "inline", soit dans un nouvel onglet du navigateur, offrant ainsi une largeur accrue).

Deux largeurs prédéterminées sont possibles : 1024px ou 1900px.

Le choix entre ces deux valeurs est déterminé par le contenu du champ "linkType", présent dans la table MySQL "_menu".

Seules deux valeurs de champ sont déterminantes, pour un affichage du tableau dans un nouvel onglet, d'une largeur de 1900px : "uriTable" et "uriPrint".

Valeur retournée par la fonction

Anonyme SA

- Aucune

```
IX. public function setColWidth($col, $width)
```

Détermine la largeur d'une colonne à afficher dans une table.

Variables

\$col	Nom de la colonne	Chaîne de caractères
\$width	Largeur de la colonne \$col	Entier numérique

Valeur retournée par la fonction

- Aucune

```
X. public function setColsWidths()
```

Utilise la propriété de la classe "`_PGEParams['tableColsWidths']`", qui détermine la largeur de plusieurs colonnes à afficher. Cette fonction reprend l'usage de la fonction précédente "`setColWidth`", dans une boucle d'itération.

Exemple :

```
$gestionnairePHPGrid -> _PGEParams['tableColsWidths']['Cols'] =  
'NameCol1,Namecol2,Namecol3'
```

```
$gestionnairePHPGrid -> _PGEParams['tableColsWidths']['Widths'] = '50,150,300';
```

Valeur retournée par la fonction

- Aucune

```
XI. public function prepareStatements()
```

Prépare l'affichage du tableau concerné, selon les propriétés attribuées à la classe.

La source des divers paramètres est constituée par la propriété "`$_PGEParams`", dont plusieurs valeurs sont inscrites dans le fichier PHP en charge de l'affichage du tableau demandé.

Valeur retournée par la fonction

- Aucune

```
XII. public function executeStatements()
```

Attribue divers paramètres au tableau à afficher, provenant de la table MySQL "`_grid_definers`", où plusieurs propriétés sont définies.

Valeur retournée par la fonction

- Aucune

I. private function checkUsersVsGroup(\$groups)

Obtention des comptes utilisateurs selon un groupe donné, en vue de construire des listes à options pour les formulaires d'ajout d'enregistrement "PHPGrid".

Variable

\$groups	Nom de groupe	Chaîne de caractères
-----------------	---------------	----------------------

Valeur retournée par la fonction

- Retourne un tableau contenant les utilisateurs appartenant au groupe "\$groups", ou une chaîne de caractères vide, si aucune valeur n'est trouvée.

II. private function getValuesFromDefinersForMandatoryFields(\$table)

Récupère des valeurs de la table "_grid_definers" déterminant les champs obligatoires à remplir d'un formulaire "PHPGrid".

Variable

\$table	Nom d'une table	Chaîne de caractères
----------------	-----------------	----------------------

Valeur retournée par la fonction

- Retourne un tableau contenant les noms des champs obligatoires.

III. private function getValuesFromDefinersForSelectFields(\$table, \$field)

Récupère la valeur du champ "sourcecode" dans la table "_grid_definers", en vue de construire une liste à options.

Le champ "sourcecode" contient des valeurs à inscrire dans la liste à option selon le format suivant:

valeur:nom de valeur;

Exemple avec des durées en minutes :

::15:15 min;30:30 min;60:1h;120:2h;240:4h;480:8h;

La première entrée < **::;** > spécifie une entrée sans valeur dans la liste à options.

Variables

\$table	Nom d'une table	Chaîne de caractères
\$field	Nom du champ de la table	Chaîne de caractères

Valeur retournée par la fonction

- Retourne une chaîne de caractères spécifiant les options à disposition pour la liste.

Fonctions liées à des tables particulières

```
I. private function getAssetsInventoryNumbersForSelect()
```

Récupère le numéro d'inventaire de la table "d_assets"

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

```
II. private function getUsersFNForSelect()
```

Récupère les nom et prénom des utilisateurs en vue de construire des listes d'options HTML.

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

```
III. private function getUsersUNForSelect()
```

Récupère le nom d'utilisateur des utilisateurs en vue de construire des listes d'options HTML.

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

```
IV. private function getUsersUNVsGroupForSelect($group = "")
```

Récupère le nom d'utilisateur des utilisateurs selon un groupe donné, en vue de construire des listes d'options HTML.

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

```
V. private function getGroupsNForSelect()
```

Récupère le nom des groupes en vue de construire une liste à options.

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

VI. private function getMaterialsForSelect()

Récupère les noms de types de matériels (p. ex. "Serveur", "Clef USB", etc.) de la table "s_materials", en vue de construire une liste à options.

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

VII. private function getAddressesDataForSelect(\$field)

Récupère le un champ donné "\$field" de la table "s_addresses" (Adresse des bureaux de l'entreprise "Anonyme SA"), en vue de construire une liste à options.

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

VIII. private function getCategoriesForSelect(\$table, \$field)

Récupère un nom de type selon la table "\$table" et le champ "\$field" de la table "s_categories" (p. ex. pour la table "d_assets" et pour le champ "OS", une liste des Systèmes d'exploitation les plus courants), en vue de construire une liste à options.

Variables

\$table	Nom d'une table	Chaîne de caractères
\$field	Nom du champ de la table	Chaîne de caractères

Valeur retournée par la fonction

- Retourne une chaîne de caractères au format "valeur:nom de valeur;", spécifiant les options à disposition pour la liste.

Fonctions liées à des pseudo-événements "jQuery"

Le principe de ces fonctions consiste en l'adjonction de code JavaScript à la section courante du site, selon les besoins.

```
I. public function onBeforeShowForm_SetDateModificationRecord()
```

Récupère la date du jour, en vue de l'attribuer au champ "date de modification", d'un formulaire "PHPGrid", avant l'affichage du formulaire.

Evènement

<code>onBeforeShowForm</code>	Avant l'affichage d'un formulaire "PHPGrid"
-------------------------------	---

Valeur retournée par la fonction

- Aucune

```
II. public function onBeforeShowForm_SetUserModificationRecord($user)
```

Récupère le nom de l'utilisateur courant, en vue de l'attribuer au champ "Modifié par", d'un formulaire "PHPGrid", avant l'affichage du formulaire.

Evènement

<code>onBeforeShowForm</code>	Avant l'affichage d'un formulaire "PHPGrid"
-------------------------------	---

Variable

<code>\$user</code>	Utilisateur courant	Chaîne de caractères
---------------------	---------------------	----------------------

Valeur retournée par la fonction

- Aucune

```
III. public function onBeforeShowForm_SetUserCreationRecord($user)
```

Affecte le nom d'utilisateur courant au champ "Créé par", un formulaire "PHPGrid" avant l'affichage du formulaire.

Evènement

<code>onBeforeShowForm</code>	Avant l'affichage d'un formulaire "PHPGrid"
-------------------------------	---

Variable

<code>\$user</code>	Utilisateur courant	Chaîne de caractères
---------------------	---------------------	----------------------

Valeur retournée par la fonction

- Aucune

```
IV. public function onBeforeShowForm_DeleteRecord()
```

Récupère l'identifiant de la ligne à supprimer, avant l'affichage du formulaire.

Evènement

onBeforeShowForm	Avant l'affichage d'un formulaire "PHPGrid"
-------------------------	---

Valeur retournée par la fonction

- Aucune

```
V. public function onEditAfterShowForm($col)
```

Affecte l'attribut "lecture seule" d'une colonne "\$col" après l'affichage du formulaire.

Evènement

onEditAfterShowForm	Durant l'édition d'un formulaire "PHPGrid", après l'affichage de celui-ci.
----------------------------	--

Variable

\$col	Nom de la colonne d'une table	Chaîne de caractères
--------------	-------------------------------	----------------------

Valeur retournée par la fonction

- Aucune

```
VI. public function onAfterDeleteRecord($section, $url)
```

Recompose l'URL du formulaire après de la suppression d'un enregistrement, en ajoutant des composantes indiquant la suppression d'un enregistrement, ainsi que son identifiant.

Evènement

onAfterDeleteRecord	Après la suppression d'un enregistrement
----------------------------	--

Variables

\$section	Nom de la section courante du site	Chaîne de caractères
\$url	Le lien de base du site	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

VII. public function onBeforeSubmitRecord()

Recompose l'URL du formulaire avant sa soumission, en ajoutant une composante indiquant l'ajout ou la modification d'un enregistrement.

Evènement

onBeforeSubmitRecord	Avant la soumission d'un nouvel enregistrement ou sa modification.
-----------------------------	--

Valeur retournée par la fonction

- Aucune

VIII. public function onAfterSubmitRecordWithMail(\$section, \$url)

Recompose l'URL du formulaire après sa soumission, en ajoutant une composante indiquant l'ajout ou la modification d'un enregistrement en vue de l'envoi d'un courriel.

Evènement

onAfterSubmitRecord	Après l'ajout ou la modification d'un enregistrement.
----------------------------	---

Variables

\$section	Nom de la section courant du site	Chaîne de caractères
\$url	Le lien de base du site	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

IX. public function onAfterSubmitRecordNew(\$section, \$url)

Recompose l'URL du formulaire après sa soumission, en ajoutant des composantes indiquant l'ajout d'un enregistrement, ainsi que l'identifiant de l'enregistrement concerné. L'identifiant est de valeur "Null" dans le cas d'un ajout.

Evènement

onAfterSubmitRecord	Après l'ajout ou la modification d'un enregistrement.
----------------------------	---

Variables

\$section	Nom de la section courant du site	Chaîne de caractères
\$url	Le lien de base du site	Chaîne de caractères

Valeur retournée par la fonction

- Aucune

```
X. public function onBeforeScriptEnd_MultiExport($path, $csv = 0, $excel = 0, $html = 0, $pdf = 0)
```

Avant la fin du script "jqGrid", recompose la barre d'outils d'export, selon les variables données.

Evènement

onBeforeScriptEnd	Avant la fin d'exécution du script "jqGrid"
--------------------------	---

Variables

\$path	Nom de la section courant du site	Chaîne de caractères
\$csv	Facultatif Type de fichier "CSV", disponible pour l'export <u>Valeurs disponibles :</u> Par défaut false	Booléen
\$excel	Facultatif Type de fichier "XLS", disponible pour l'export <u>Valeurs disponibles :</u> Par défaut false	Booléen
\$html	Facultatif Type de fichier "HTML", disponible pour l'export <u>Valeurs disponibles :</u> Par défaut false	Booléen
\$pdf	Facultatif Type de fichier "PDF", disponible pour l'export <u>Valeurs disponibles :</u> Par défaut false	Booléen

Valeur retournée par la fonction

- Aucune

```
XI. public function onAfterEditShowForm_PermitAddWithReadOnlyFields($aFields)
```

Prépare le formulaire pour la soumission d'un enregistrement, en affectant aux champs en lecture seule, la possibilité d'écriture. Sans cela l'enregistrement ne peut être écrit correctement.

Evènement

<code>onAfterEditShowForm</code>	Après l'affichage du formulaire d'édition
----------------------------------	---

Variable

<code>\$aFields</code>	Tableau des champs de la table à traiter	Tableau
------------------------	--	---------

Valeur retournée par la fonction

- Aucune

Fonctions "utilitaires" de la classe

```
XII. public function getMenuLabelFromSection($section)
```

Recherche le libellé du menu d'une section du site.

Variable

<code>\$section</code>	Section du site	Chaîne de caractères
------------------------	-----------------	----------------------

Valeur retournée par la fonction

- Chaîne de caractères représentant la section du site recherchée.

```
XIII. public function manageTableRowAttachment($id, $op, $absURL, $absURI)
```

Gère l'affectation d'une pièce jointe à l'enregistrement d'une table.

"PHPGrid" dispose d'une fonction d'ajout de fichier à un enregistrement de table. Le but unique étant d'afficher des images dans la colonne d'un tableau.

A cette fin, le module ne traite qu'avec des chemins internes du serveur (p. ex. "/var/html/site/pub/image.jpg"), ce qui ne satisfait pas au besoin d'adjoindre d'autres types de fichiers (pdf, docx, etc.).

C'est pourquoi, la présente fonction gère les liens en substituant le lien interne du serveur par un lien "HTML" et affecte une icône dans une colonne d'un tableau prévue à cet effet. Cela permet de signifier à l'utilisateur qu'un fichier est adjoint à un enregistrement. Le fichier est ensuite accessible en ouvrant l'enregistrement concerné.

Une fois un fichier adjoint à un enregistrement, il est nécessaire de rafraîchir la page pour visualiser l'icône.

Variables

<code>\$id</code>	Identifiant de l'enregistrement	Entier numérique
<code>\$op</code>	Opération effectuée Valeurs traitées : <code>mod</code> (modification) <code>del</code> (delete) <code>add</code> (add)	Chaîne de caractères
<code>\$absURL</code>	Hyperlien absolu	Chaîne de caractères
<code>\$absURI</code>	Hyperlien relatif	Chaîne de caractères

Valeur retournée par la fonction

- "true" si l'opération a réussi ou "false" en cas d'échec.

```
XIV. public function getTableRowAttachment($id)
```

Recherche le nom de fichier de la pièce jointe de l'enregistrement d'une table. Utilisé par la fonction "manageTableRowAttachment".

Variable

<code>\$id</code>	Identifiant de l'enregistrement	Entier numérique
-------------------	---------------------------------	------------------

Valeur retournée par la fonction

- Le nom du chemin du fichier ou un chaîne vide.

```
XV. public function getTableRowAttachmentExt($id)
```

Recherche l'extension de fichier de la pièce jointe de l'enregistrement d'une table.

Variable

<code>\$id</code>	Identifiant de l'enregistrement	Entier numérique
-------------------	---------------------------------	------------------

Valeur retournée par la fonction

- L'extension du fichier ou un chaîne vide.

```
XVI. public function setTableRowAttachmentLink($id, $file, $absURL, $absURI)
```

Affecte un hyperlien absolu à la balise de code "HTML" `` d'un fichier, joint à un enregistrement.

Variables

<code>\$id</code>	Identifiant de l'enregistrement	Entier numérique
<code>\$op</code>	Opération effectuée <u>Valeurs traitées :</u> <code>mod</code> (modification) <code>del</code> (delete) <code>add</code> (add)	Chaîne de caractères
<code>\$absURL</code>	Hyperlien absolu	Chaîne de caractères
<code>\$absURI</code>	Hyperlien relatif	Chaîne de caractères

Valeur retournée par la fonction

- "true" si l'opération a réussi ou "false" en cas d'échec.

```
XVII. public function setTableRowAttachmentIcon($id,$ext)
```

Affecte une icône en regard du type de pièce jointe à l'enregistrement d'une table.

Variables

<code>\$id</code>	Identifiant de l'enregistrement	Entier numérique
<code>\$ext</code>	Extension du fichier à traiter	Chaîne de caractères

Valeur retournée par la fonction

- "true" si l'opération a réussi ou "false" en cas d'échec.

```
XVIII. public function unsetTableRowAttachmentIcon($id)
```

Désaffecte une icône liée à une pièce jointe, lors de la suppression de ladite pièce jointe d'un enregistrement.

Variable

<code>\$id</code>	Identifiant de l'enregistrement	Entier numérique
-------------------	---------------------------------	------------------

Valeur retournée par la fonction

- "true" si l'opération a réussi ou "false" en cas d'échec.

Table " grids definers"

Cette table étant fortement liée à la classe "PHPGridHandle", le choix a été fait de la présenter en tant que sous-chapitre de cette classe PHP.

La table permet le stockage de plusieurs paramètres relativement aux tableaux d'une section du site Web, tels que la largeur d'un champ de saisie dans un formulaire, si un champ est affiché dans le tableau de la section du site Web, etc.

Nombre de fonctions PHP de la présente classe font appel à cette table.

Description des champs de la table " grids definers"

Nom	Commentaire
Id	Identifiant unique
bench	Nom d'un table MySQL
viewType	Nom d'une vue
fieldName	Nom du champ de la table
fieldLabel	Libellé du champ
fieldType	Type de champ Valeurs possibles : integer : Entier numérique text : Chaîne de caractères textarea : Chaîne de caractères lignes multiples select : Liste de choix checkbox : Boîte à cocher currency : Numéraire number : Nombre à virgule
source	Type de source pour un élément "select" Valeurs possibles :

Nom	Commentaire
	dbarray : Une table MySQL comme source innerarray : Un tableau PHP phparray : Code présent dans la colonne "sourcecode"
sourcecode	<p>Dans le cas d'un élément de type "select", désigne des valeurs mises à disposition, selon le format : <valeur>:<libellé>; Ex: ;;<val1>:<libellé1>;<val2>:<libellé2></p> <p>La mention ":;" permet la possibilité de sélectionner une valeur vide</p>
mandatory	<u>Formulaire de saisie</u> : Champ obligatoire
readonly	<u>Formulaire de saisie</u> : Champ en lecture seule
tableView	Champ visible dans le tableau représenté sur la page du site Web
formView	<u>Formulaire de saisie</u> : Champ visible dans le formulaire de saisie
formHeader	<u>Formulaire de saisie</u> : Titre affiché au-dessus du champ de saisie
formTip	<u>Formulaire de saisie</u> : Texte d'une astuce à afficher pour un champ de saisie (S'affiche sous la forme d'une icône "?")
formRowPos	<u>Formulaire de saisie</u> : Position horizontale du champ de saisie
formColPos	<u>Formulaire de saisie</u> : Position verticale du champ de saisie (on peut aligner des champs de saisie sur deux colonnes, p. ex.)
formFunctionCall	<u>Formulaire de saisie</u> : Entier numérique désignant une fonction PHP à appeler pour compléter des champs de saisie telles que des listes d'options.
formInsideFunctionCall	<u>Formulaire de saisie</u> : Entier numérique désignant une fonction à appeler pour compléter des champs de saisie telles que des listes d'options (en lien avec la colonne formFunctionCall).
fieldMultiSelect	<u>Formulaire de saisie</u> : Booléen permettant des sélections multiples dans une liste d'options. (Buggé)
fieldEditOptions	<u>Formulaire de saisie</u> : Permet par exemple de définir la largeur d'un champ de saisie.

Classe "security.class.php"

Ensemble de fonctions relatives à la sécurité, tel que les contrôles de variables transmises depuis un formulaire, l'existence de fichiers, etc.

Propriétés de classe

```
- public static $errXSS = "XSS";
```

Type:	Chaîne de caractères
Portée:	Publique
Description:	Type d'erreur "XSS" (cross-site scripting)
Contenu par défaut:	"XSS"

```
- public static $errHJCK = "HIGHJACKING";
```

Type:	Chaîne de caractères
Portée:	Publique
Description:	Type d'erreur de l'ordre du détournement
Contenu par défaut:	"HIGHJACKING"

```
- public static $errGEN = "GENERAL";
```

Type:	Chaîne de caractères
Portée:	Publique
Description:	Type d'erreur d'ordre général
Contenu par défaut:	"GENERAL"

```
- public static $urlRedirect = "";
```

Type:	Tableau
Portée:	Publique
Description:	URL de redirection de page lorsqu'une erreur survient
Contenu par défaut:	Chaîne vide

Anonyme SA

- private static \$errorPage

Type:	Tableau
Portée:	Privée
Description:	Codes d'erreur associés à une page PHP
Contenu par défaut:	<pre>array (400 => "400.php", 401 => "401.php", 403 => "403.php", 404 => "404.php", 405 => "405.php", 406 => "406.php", 407 => "407.php", 408 => "408.php", 500 => "500.php", 502 => "501.php", 503 => "503.php", 504 => "504.php", 511 => "511.php", 800 => "800.php", 801 => "801.php", 802 => "802.php", 810 => "810.php", 811 => "811.php", 815 => "815.php", 816 => "816.php", 817 => "817.php", 820 => "820.php", 830 => "830.php", 840 => "840.php", 900 => "900.php", 901 => "901.php", 999 => "999.php")</pre>

Méthodes de classe

```
I. public function setEncoding($str, $newEncoding)
```

Retourne la conversion d'une chaîne de caractères depuis l'encodage de caractères courant en un nouvel encodage (p.ex. "UTF-8" vers "ISO-8859-1").

Variables

<code>\$str</code>	Chaîne de caractères à convertir	Chaîne de caractères
<code>\$newEncoding</code>	Nouvel encodage	Chaîne de caractères

Valeur retournée par la fonction

- Chaîne de caractères convertie dans le nouvel encode de caractères

```
II. public function blacklistFilter($str)
```

Filtre une chaîne de caractères selon une expression régulière "`< \s \c \r \i \p \t >`". La fonction retourne la chaîne ou redirige vers une page d'erreur.

Variable

<code>\$str</code>	Chaîne de caractères à filtrer	Chaîne de caractères
--------------------	--------------------------------	----------------------

Valeur retournée par la fonction

- Chaîne de caractères ou redirection vers une page d'erreur

```
III. public function whitelistFilter($str, $whiteFilterPattern = "", $errCode = 0, $min
    = 8, $max= 68, $urlRedirect = "")
```

Contrôle une chaîne de caractères selon un choix d'expressions régulières, exprimé par la variable "\$whiteFilterPattern".

Variables

\$str	Chaîne de caractères à contrôler	Chaîne de caractères
\$whiteFilterPattern	Facultatif Types d'expressions régulières à utiliser pour la vérification de la chaîne Valeurs possibles : Valeur par défaut Chaîne vide String ([a-zA-Z]+) Number /^([0-9]+)\$/ Numbers /^(\d{1,10})\$/ Password /^((?=.*\d) (?=.*[A-Za-z]) (?=.*[A-Z]) (?=.*[a-z]) (?=.*[!#\$%&éèàçéäöü°\$ '\"(\) * +,-.\[/\\]^ _`{ }~\") [0-9A-Za-z!#\$%&éèàçéäöü°\$ '\"(\) * +,-.\[/\\]^ _`{ }~\"){".\$min.", ".\$max."})\$/i Username /([A-Za-z0-9_@-]+)\$/i Everything /^(.*)\$/i Alphanumeric ([0-9a-zA-Z]+) passwordConstraints /^((?=.*\d) (?=.*[a-z]) (?=.*[A-Z]) (?=.*[!@#\$%&*()_+~\-=\[\]{};:~.?]) (?=.*[a-zA-Z]) .{8,30})\$/	Chaîne de caractères

\$errCode	(En développement) Facultatif Type d'erreur rapporté <u>Valeurs possibles :</u> Valeur par défaut 0	Entier numérique
\$min	Facultatif Nombre minimum de caractères dans la chaîne à analyser <u>Valeurs possibles :</u> Valeur par défaut 8 Nombre entier	Entier numérique
\$max	Facultatif Nombre de caractères maximal <u>Valeurs possibles :</u> Valeur par défaut 68 Nombre entier	Entier numérique
\$urlRedirect	Facultatif Lien de redirection vers une page d'erreur <u>Valeurs possibles :</u> Valeur par défaut Chaîne vide Chaîne de caractères	Chaîne de caractères

Valeur retournée par la fonction

- Chaîne de caractères ou une chaîne vide

IV. public function riseSecError (\$eCode)

Redirige vers une page affichant une erreur de sécurité. Cette fonction n'est valable que lorsque les en-têtes HTML n'ont pas encore été injectés.

Variable

\$eCode	Code d'erreur (Voir tableau \$errorPage pour les codes)	Entier numérique
----------------	--	------------------

Valeur retournée par la fonction

- Aucune

V. public function riseSecErrorAfterHeaders (\$eCode)

Redirige vers une page affichant une erreur de sécurité, après injection des en-têtes "HTML".

Variable

\$eCode	Code d'erreur (Voir tableau \$errorPage pour les codes)	Entier numérique
----------------	--	------------------

Valeur retournée par la fonction

- Aucune

VI. public function checkBrowserAndIP(\$ipTVF, \$UATVF, \$errCode = 999)

Contrôle de l'adresse IP et de l'empreinte du navigateur du client, afin de se prémunir contre la technique d'usurpation ("spoofing").

Variables

\$ipTVF	Adresse IP de client	Chaîne de caractères
\$UATVF	Navigateur du client	Chaîne de caractères
\$errCode	Facultatif Code d'erreur (Voir tableau \$errorPage pour les codes) <u>Valeurs possibles :</u> Valeur par défaut 999 Nombre entier	Entier numérique

Valeur retournée par la fonction

- Retourne "true" si le contrôle est négatif, "false" si l'usurpation est avérée.

```
VII. public function execIfFileExists ($pathToFile)
```

Vérifie l'existence d'un fichier selon le chemin complet "\$pathToFile".

Variable

\$pathToFile	Chemin complet d'un fichier	Chaîne de caractères
---------------------	-----------------------------	----------------------

Valeur retournée par la fonction

- Retourne une chaîne de caractères représentant le chemin complet du fichier en cas de succès, ou redirige vers une page d'erreur "404" (File not found).

```
VIII. public function decryptAsJSSimpleDecrypt ($str)
```

Décode une chaîne de caractères encodée.

La fonction est utilisée en combinaison avec des fonctions JavaScript pour des traitements de variables de formulaires, p.ex.

Variable

\$str	Chaîne de caractères à décoder Les caractères de la chaîne sont représentés par leur nombre ascii augmenté de 10. Chaque caractère encodé est séparé par un tiret. (Ne pas utiliser cette fonction pour des variables d'ordre confidentielles telles que des mots de passe)	Chaîne de caractères
--------------	--	----------------------

Valeur retournée par la fonction

- Retourne la chaîne de caractères décodée.

Classe "session.class.php"

Traitement de sessions PHP, utilisées sur le site. Les sessions PHP peuvent être comparées aux fichiers "cookies", permettant de stocker certaines variables lors de la visite d'un site.

Méthodes de classe

```
I. public function preventHijacking()
```

Effectue un contrôle de l'adresse IP ainsi que de l'agent du navigateur, utilisés par le client afin d'éviter une usurpation. Même fonction que la fonction "[checkBrowserAndIP](#)", mais en utilisant des variables de session.

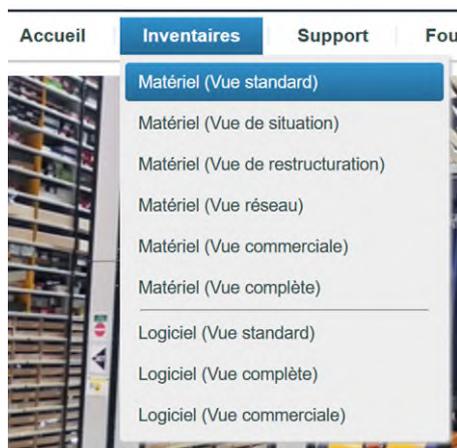
Valeur retournée par la fonction

- Retourne "true" si le contrôle est négatif, "false" si l'usurpation est avérée.

Annexes

Annexe I. Affichage du tableau "d_assets" selon la vue "standard"

Exemple de l'affichage de la table "d_assets", lors de l'appel de la section du site "Inventaire" - "Matériel (vue standard)".



Principe

1. Lors de la sélection du menu, la section du site est appelée au moyen de la variable transmise "s" (pour section), dont la valeur est encodée en base 64.

La valeur de "s" représente le numéro de section du site, présent dans la table "_menu", champ "section".

<https://nomdusite/?s=MTAxMA==> (dans notre exemple, "s" vaut "1010" encodé en base 64).

2. Suite à cet appel, la page demandée est affichée et présente le tableau d'inventaire et ses enregistrements.
3. On peut ajouter, modifier ou supprimer un enregistrement au moyen des icônes de gestion du tableau (fonctions internes au module "PHPGrid"). Des barre d'outils intégrant ces fonctions sont situées au-dessus ET en-dessous du tableau.

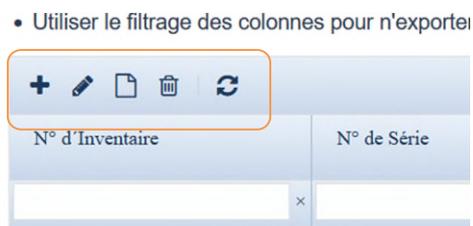


Figure 13: Icônes de gestion du tableau

4. La barre d'outils située au bas du tableau contient, outre les icônes mentionnées ci-dessus, une icône d'export. Celle-ci permet l'export des enregistrements présentés sur la page en cours, selon les colonnes affichées dans un format de fichier tel que "PDF" ou "XLS".



5. Lors d'opérations telles que mentionnées au point 3 de ce chapitre, l'identifiant de l'enregistrement concerné est récupéré au moyen de JavaScript, selon un évènement "JQGrid" tel que "OnAfterSubmitRecord".
6. L'identifiant est ensuite injecté dans un lien sous la forme "https://nomdusite/?s=MTAxMA==&ae=ODky", au moyen de JavaScript.

"ae" représente l'enregistrement de la table sur lequel on a travaillé.
7. Toujours via JavaScript, la page PHP est ensuite rechargée avec, cette fois, la variable "ae" intégrée à l'URL.
8. Lors d'une suppression d'enregistrement, une variable "del" est ajoutée à l'URL, pour caractériser la suppression.
9. Ainsi reprise, la variable "ae" est utilisée afin de récupérer des informations de la table "d_assets" pour l'enregistrement concerné, qui sont ensuite envoyées par E-mail aux utilisateurs.
10. Ce courriel indique quelle opération a été effectuée (ajout, suppression, modification); Qui a exécuté l'opération, ainsi que les détails de l'enregistrement concerné.



Figure 14: E-mail reçu suite à la modification d'un enregistrement dans la table "d_assets"

Schéma bloc

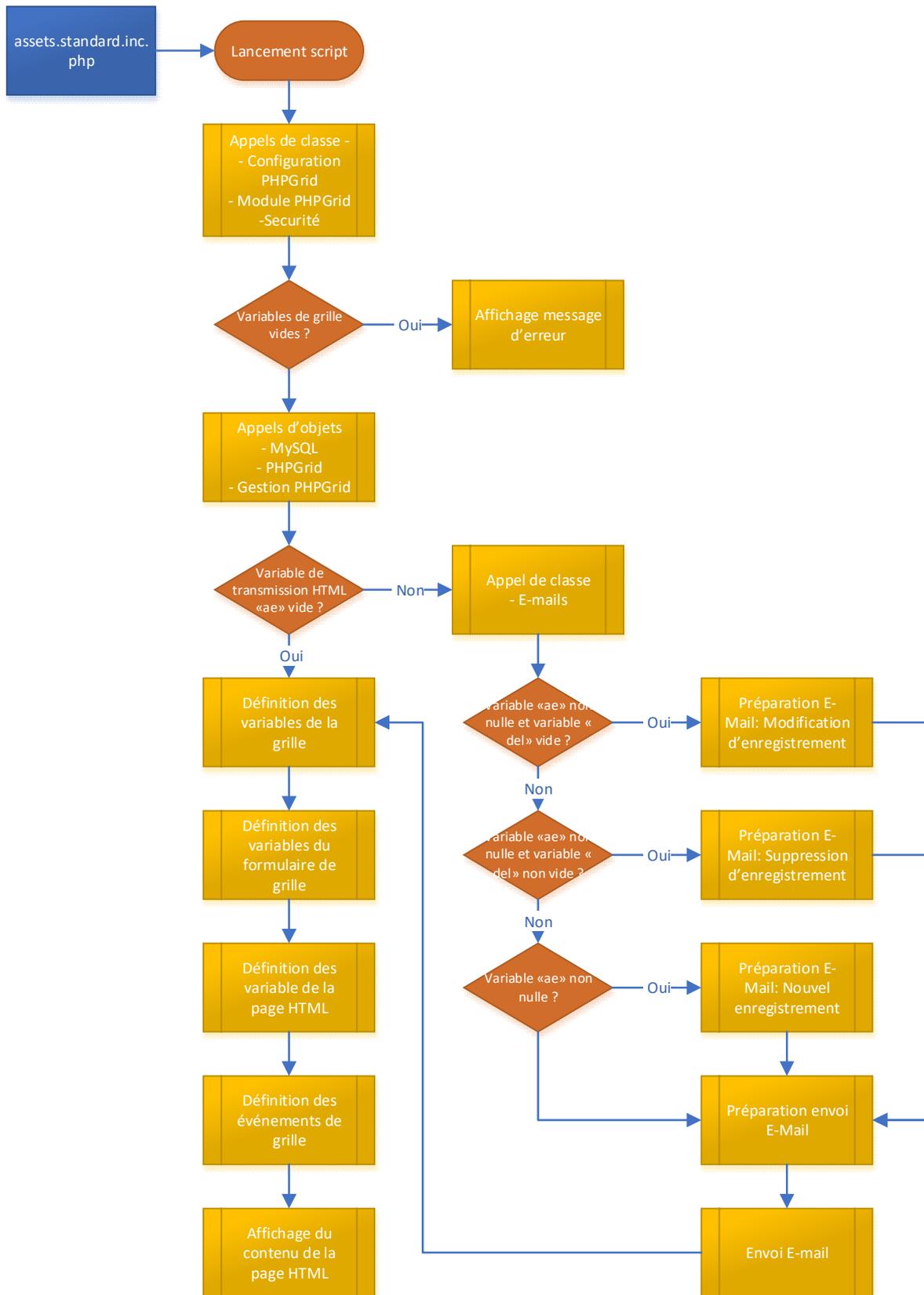


Figure 15: Schéma bloc "assets.standard.inc.php"

Code "typique" d'une page de section pour une utilisation avec "PHPGrid".

Cet exemple est tiré d'une section du site en charge d'afficher un tableau de l'inventaire des matériels actifs possédés par "Anonyme SA".

Table MySQL des données source : "d_assets"

Commentaire : Contient des matériels d'inventaire (Postes de travail, Routeurs, Commutateurs réseau, Etiqueteuse, UPS, etc.)

Légende du code PHP

Code PHP	\$pgh_hdl -> onBeforeShowForm_SetDateModificationRecord();
Mots clefs PHP	require_once
Chaînes de caractères	"Ceci est une chaîne"
Chaînes de caractères	'Ceci est une chaîne'
Constantes	ThisIsAConstant
Commentaires	// Ceci est un commentaire
Nombres	820

Code

```
<?php
/*****
 * PHP Language Version 7.4.x
 *
 * @version v1
 *
 * @creation 2021/05/10
 *
 * @copyright 2021 Anonyme SA
 *
 * @Author Fond-Fonfon
 *
 * @update 2021/10/26
 *
 *****/
?>
    <div id="contentMain" class="contentMain">
<?php

// Appel des fichiers du module "PHPGrid"
require_once (dirPLG . "phpgrid/conf.php");

require_once (dirCLS . "phpgrid.class.php");

// Création d'un nouveau gestionnaire "CheckSecurity"
$sec_hdl = new checkSecurity();

// Test si les variables globales disposent d'une valeur, sinon renvoi et
// affichage d'une erreur
if ($GLOBALS['gridACR'] == null || $GLOBALS['gridName'] == null || $GLOBALS['gridView'] == null) {

    $sec_hdl::riseSecErrorAfterHeaders(820);

}

// Champs de la table d'inventaire matériel "d_assets" à considérer dans
// plusieurs fonctionnalités
$ACTUALFields = "invNumber,serial,brand,model,acqProvider,acqDate,acqPrice,offGroup,offUser,netHost";

// Création d'un nouveau gestionnaire "MySQL"
$db_hdl = new sqlConnHandle($GLOBALS['$_MYSQL']);

// Création d'un nouveau gestionnaire "PHPGrid"
$pg_hdl = new phpCtrl\C_DataGrid("SELECT * FROM " . $GLOBALS['gridName'], "id", $GLOBALS['gridName']);

// Création d'un nouveau gestionnaire "PHPGridHandle"
$pgh_hdl = new phpGridHandle($GLOBALS['gridName'],$GLOBALS['gridView'],$db_hdl,$pg_hdl);

// Test si la variable de site "ae" existe.
// La variable contient un identifiant d'enregistrement de table,
// lors d'une opération sur celui-ci (ajout/suppression/modification)
if (isset($_GET['ae'])) {

    // Création d'un nouveau gestionnaire "MySQL"
    $mail_hdl = new MailManager($db_hdl);

    // Test si "ae" ne vaut pas "Null" ni "del"
    if (base64_decode($_GET['ae']) != 'null' && !(isset($_GET['del']))) {

        // Retour "ae" de la variable après vérification par expression
        // régulière
        $ae = $sec_hdl -> whitelistFilter(base64_decode($_GET['ae']),
            "numbers",0,8,68,urlSITE);

        // Attribution de deux valeurs utilisé dans le corps d'un mail
        // à envoyer
        $mail_hdl -> sectionParams['subTitle'] = "Modification d'un Enregistrement";

        $mail_hdl -> mailParams['subject'] = siteCompany . " - " . "Inventaire materiel - Modification";

    }

    // Test si "ae" ne vaut pas "Null" et si la variable "del" est attribuée
    // "del" contient un un enregistrement supprimé dans la table encodé
```

```

// "base64"
elseif (base64_decode($_GET['ae']) != 'null' && isset($_GET['del'])) {

    // Retour "ae" de la variable après vérification par expression
    // régulière
    $ae = $sec_hdl -> whitelistFilter(base64_decode($_GET['ae']),
        "numbers",0,8,68,urlSITE);

    // Supprime les tags HTML et décode "del"
    $mail_hdl -> tableParams['onDeleteString'] = strip_tags(base64_decode($_GET['del']));

    // Attribution de deux valeurs utilisé dans le corps d'un mail
    // à envoyer
    $mail_hdl -> sectionParams['subTitle'] = "Suppression d'un Enregistrement";

    $mail_hdl -> mailParams['subject'] = siteCompany . " - " . "Inventaire materiel - Suppression";

}

// Test si "ae" vaut "Null"
elseif (base64_decode($_GET['ae']) == 'null') {

    // Attribution de deux valeurs utilisé dans le corps d'un mail
    // à envoyer
    $mail_hdl -> sectionParams['subTitle'] = "Ajout d'un Enregistrement";

    $mail_hdl -> mailParams['subject'] = siteCompany . " - " . "Inventaire materiel - Ajout";

    // Recherche le dernier enregistrement ajouté dans la table
    $query = "SELECT MAX(id) FROM " . $GLOBALS['gridName'] . ";";

    if ($result = $db_hdl->SQLHandle->query($query)) {

        if ($row = $result->fetch_assoc()) {

            // Attribue l'identifiant du dernier enregistrement à $ae
            $ae = $row['MAX(id)'];

        }

    }

}

// Attribution de plusieurs valeurs utilisées dans le corps d'un mail
// à envoyer
$mail_hdl -> sectionParams['title'] = "Inventaire matériel";
$mail_hdl -> sectionParams['object'] = "Enregistrement " . $ae;

$mail_hdl -> tableParams['table'] = $GLOBALS['gridName'];
$mail_hdl -> tableParams['fields'] = $ACTUALFields;
$mail_hdl -> tableParams['viewType'] = $GLOBALS['gridView'];
$mail_hdl -> tableParams['recordID'] = $ae;

$mail_hdl -> mailParams['to'] = mailDestination;
$mail_hdl -> mailParams['headers'][] = "From: noreply@anonyme.ch";
$mail_hdl -> mailParams['headers'][] = "Reply-To: webmaster@anonyme.ch";
$mail_hdl -> mailParams['headers'][] = "MIME-Version: 1.0";
$mail_hdl -> mailParams['headers'][] = "Content-Type: text/html; charset=utf8";
$mail_hdl -> mailParams['headers'][] = "X-Mailer: PHP/" . phpversion();
$mail_hdl -> mailParams['signature'] = mailSignature;
$mail_hdl -> mailParams['company'] = siteCompany;
$mail_hdl -> mailParams['dpt'] = siteDepartment;

if ($ae != "") {

    // Envoi de l'E-mail en cas d'ajout, de modification ou de suppression
    // d'enregistrement
    $send = $mail_hdl -> sendMail("phpGrid");

}

}

// Attribution d'un filtre à une requête "PHP" effectuée par "PHPGrid"
$spgh_hdl -> _PGEParams['queryFilter'] = "status=1";

// Attribution d'un filtre de tri à une requête "PHP" effectuée par "PHPGrid"
$spgh_hdl -> _PGEParams['tableSort']['field'] = "acqDate";
$spgh_hdl -> _PGEParams['tableSort']['direction'] = "DESC";

// Attribution de la largeur graphique du tableau à afficher.
$spgh_hdl -> _PGEParams['tableWidth'] = $spgh_hdl->setGridSizeFromMenuViewStyle($GLOBALS['section']);
// Ne pas afficher le libellé du tableau.
$spgh_hdl -> _PGEParams['tableShowCaption'] = false;
// Afficher la barre d'outil du tableau au sommet de ce dernier.
$spgh_hdl -> _PGEParams['tableSetToolbarOnTop'] = true;
// Format d'export disponible du tableau
$spgh_hdl -> _PGEParams['tableExport']['format'] = "EXCEL";
// Autoriser la recherche d'éléments dans le tableau
$spgh_hdl -> _PGEParams['tableEnableSearch'] = true;
// Regroupement des éléments affichés selon la colonne "OffCity"
$spgh_hdl -> _PGEParams['tableGroupBy'] = "offCity";

// Largeur graphique du formulaire lié au tableau
$spgh_hdl -> _PGEParams['formWidth'] = 900;
// Champs à remplir considérés comme obligatoire
$spgh_hdl -> _PGEParams['formColRequired'][0] = 'invNumber';

// Attribution de valeurs par défaut à certains champs du formulaire
$spgh_hdl -> _PGEParams['formColCreateDefValues'][0]['field'] = "dateCre";

```

```

$pg_hdl -> _PGEParams['formColCreateDefValues'][0]['value'] = date("Y-m-d");
$pg_hdl -> _PGEParams['formColCreateDefValues'][1]['field'] = "dateMod";
$pg_hdl -> _PGEParams['formColCreateDefValues'][1]['value'] = date("Y-m-d");
$pg_hdl -> _PGEParams['formColCreateDefValues'][2]['field'] = "status";
$pg_hdl -> _PGEParams['formColCreateDefValues'][2]['value'] = "1";

// Formattage du formulaire à remplir
$pg_hdl -> _PGEParams['enableEdit']['type'] = "FORM";
$pg_hdl -> _PGEParams['enableEdit']['options'] = $GLOBALS['gridACR'];

// Titres de la présente section du site Web
$pg_hdl -> _PGEParams['html']['h1'] = "Inventaire matériel";
$pg_hdl -> _PGEParams['html']['h2'] = "Vue standard";
$pg_hdl -> _PGEParams['html']['h4'] = "(Groupé par ville / Matériel actif uniquement / Tri par date d'acquisition descendant)";
$pg_hdl -> _PGEParams['html']['h5'][0]['title'] = "Edition";
$pg_hdl -> _PGEParams['html']['h5'][0]['html'][0] = "                <ul style=\"padding-left: 15px;\"";
$pg_hdl -> _PGEParams['html']['h5'][0]['html'][1] = "                <li>Double-cliquer sur un enregistrement pour modifier un
champ.</li>";
$pg_hdl -> _PGEParams['html']['h5'][0]['html'][2] = "                </ul>";
$pg_hdl -> _PGEParams['html']['h5'][1]['title'] = "Export";
$pg_hdl -> _PGEParams['html']['h5'][1]['html'][0] = "                <ul style=\"padding-left: 15px;\"";
$pg_hdl -> _PGEParams['html']['h5'][1]['html'][1] = "                <li>Seules les colonnes affichées sont exportées.</li>";
$pg_hdl -> _PGEParams['html']['h5'][1]['html'][2] = "                <li>Utiliser le filtrage des colonnes pour n'exporter que
certains enregistrements.</li>";
$pg_hdl -> _PGEParams['html']['h5'][0]['html'][3] = "                </ul>";

// Attribution d'une monnaie
$pg_hdl -> _PGEParams['currency']['monney'] = "CHF";

// Attribution des clefs à transmettre en cas de suppression d'enregistrement
$pg_hdl -> _PGEParams['keysForDelete'] = $ACTUALFields;

// Attribution des clefs de champs à transmettre en cas de suppression
// d'enregistrement
$pg_hdl -> setHtmlJQGridStyle();

// Affichage de l'environnement HTML
$pg_hdl -> openHtmlEnvironment();

// Prépare l'affichage des données de la "table d_assets"
$pg_hdl -> prepareStatements();

// Prépare l'affichage des données de la "table d_assets"
$pg_hdl -> executeStatements();

// Assigne à la table "d_assets" une largeur d'affichage graphique (JavaScript)
$pg_hdl -> setHtmlDivWidthConstraint();

// Génère un événement après soumission d'un enregistrement (JavaScript)
$pg_hdl -> onAfterSubmitRecord($GLOBALS['section'],urlSITE);

// Génère un événement avant affichage du formulaire de saisie (JavaScript)
$pg_hdl -> onBeforeShowForm_SetDateModificationRecord();

// Génère un événement avant affichage du formulaire lors d'une suppression
// d'enregistrement (JavaScript)
$pg_hdl -> onBeforeShowForm_DeleteRecord();

// Génère un événement après suppression d'un enregistrement (JavaScript)
$pg_hdl -> onAfterDeleteRecord($GLOBALS['section'],urlSITE);

// Affiche le tableau sur la page Web du site
$pg_hdl -> display();

// Affiche les balises de clôture de l'environnement HTML
$pg_hdl -> closeHtmlEnvironment();

?>
    </div>
<?php

// Affiche des champs de recherche au sommet de chaque colonne (JavaScript)
$pg_hdl -> setHtmlToggleToolbar();

?>

```

Architecture de la table "d_assets"

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
invNumber	VARCHAR(25)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
serial	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
material	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
brand	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
model	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
status	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
comment	MEDIUMTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
acqProvider	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
acqDate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
acqPrice	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
acqRent	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
offCity	VARCHAR(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
offAddress	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
offFloor	VARCHAR(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
offDesk	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
offGroup	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
offUser	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
netHost	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
netIp1	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
netIp2	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
netMac1	VARCHAR(17)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
netMac2	VARCHAR(17)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysName	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysPorts	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysPosition	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysPhone	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysCpu	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysCpuNumber	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysRam	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysDiagonal	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysOs	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysLicense	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysKey	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysEsc	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysStoptrack	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysImei	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysMeid	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysPin	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sysCode	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
dateCre	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
dateMod	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figure 16: Structure de la table "d_assets"

Légende

PK	Clef Principale (Primary Key)
NN	Ne peut être égal à "Null" (Not Null)
UQ	Index Unique (UniQue)
B	Colonne binaire (Binary column)
UN	Non signé (UNsigned)
ZF	Rempli les colonnes avec un "0" si celle-ci est numérique (Zero Fill)
AI	Incrément automatique (Auto Increment)
G	Colonne générée (Generated column)

Paramètres d'affichage de la table "d_assets" présents dans la table " grids_definers"

Paramètres appliqués à la table "d_assets" pour la vue dite "standard".

id	bench	viewType	fieldName	fieldLabel	fieldType	source	sourcecode	mandatory	readonly	tableView	formView	formHeader	formTip	formRowPos	formColPos	formFunctionCall	formInsideFunctionCall	fieldMultiSelect	fieldEditOptions
81	d_assets	standard	id	N° Identifiant	integer	NULL	NULL	0	0	0	0	Données d...		1	2	NULL	NULL	0	NULL
82	d_assets	standard	invNumber	N° d 'Inventaire	text	NULL	NULL	0	0	1	1			1	1	NULL	NULL	0	NULL
83	d_assets	standard	serial	N° de Série	text	NULL	NULL	0	0	1	1			2	1	NULL	NULL	0	NULL
84	d_assets	standard	material	Matériel	select	NULL	NULL	0	0	1	1			2	2	5	NULL	0	width: 95%;
85	d_assets	standard	brand	Marque	text	NULL	NULL	0	0	1	1			3	1	NULL	NULL	0	NULL
86	d_assets	standard	model	Modèle	text	NULL	NULL	0	0	1	1			3	2	NULL	NULL	0	NULL
87	d_assets	standard	status	Matériel actif	checkbox	NULL	NULL	0	0	0	1			4	1	NULL	NULL	0	NULL
88	d_assets	standard	comment	Commentaire	textarea	NULL	NULL	0	0	1	1			5	1	NULL	NULL	0	width: 95%;
89	d_assets	standard	acqProvider	Fournisseur	text	NULL	NULL	0	0	0	1	Données d...		6	1	NULL	NULL	0	NULL
90	d_assets	standard	acqDate	Date d 'acquisition	date	NULL	NULL	0	0	0	1			6	2	NULL	NULL	0	NULL
91	d_assets	standard	acqPrice	Coût CHF (TVA ind.)	currency	NULL	NULL	0	0	0	1			7	1	NULL	NULL	0	NULL
92	d_assets	standard	acqRent	Location	checkbox	NULL	NULL	0	0	0	1			7	2	NULL	NULL	0	NULL
93	d_assets	standard	offCity	Ville	select	NULL	NULL	0	0	0	1	Situation g...		8	1	6	cityAcronym	0	width: 95%;
94	d_assets	standard	offAddress	Adresse	select	NULL	NULL	0	0	0	1			8	2	6	addressAcronym	0	width: 95%;
95	d_assets	standard	offFloor	Etage	select	NULL	NULL	0	0	0	1			9	1	6	floor	0	width: 95%;
96	d_assets	standard	offDesk	Bureau	text	NULL	NULL	0	0	0	1			9	2	NULL	NULL	0	NULL
97	d_assets	standard	offGroup	Groupe	select	NULL	NULL	0	0	0	1	Situation in...		10	1	4	NULL	0	width: 95%;
98	d_assets	standard	offUser	Utilisateur	select	NULL	NULL	0	0	0	1			10	2	3	NULL	0	NULL
99	d_assets	standard	netHost	Nom d 'hôte	text	NULL	NULL	0	0	0	1	Réseau		11	1	NULL	NULL	0	NULL
100	d_assets	standard	netip1	Adresse IP 1	text	NULL	NULL	0	0	0	1			12	1	NULL	NULL	0	NULL
101	d_assets	standard	netip2	Adresse IP 2	text	NULL	NULL	0	0	0	1			12	2	NULL	NULL	0	NULL
102	d_assets	standard	netMac1	Adresse MAC 1	text	NULL	NULL	0	0	0	1			13	1	NULL	NULL	0	NULL
103	d_assets	standard	netMac2	Adresse MAC 2	text	NULL	NULL	0	0	0	1			13	2	NULL	NULL	0	NULL
104	d_assets	standard	sysName	Nom du système	text	NULL	NULL	0	0	0	1	Système		14	1	NULL	NULL	0	NULL
105	d_assets	standard	sysPorts	Nombre de ports	integer	NULL	NULL	0	0	0	1			14	2	NULL	NULL	0	NULL
106	d_assets	standard	sysPosition	Position	text	NULL	NULL	0	0	0	1			15	1	NULL	NULL	0	NULL
107	d_assets	standard	sysPhone	Téléphone	text	NULL	NULL	0	0	0	1			15	2	NULL	NULL	0	NULL
108	d_assets	standard	sysCpu	Processeur	text	NULL	NULL	0	0	0	1	Hardware		16	1	NULL	NULL	0	NULL
109	d_assets	standard	sysCpuNu...	Nombre de cœurs	integer	NULL	NULL	0	0	0	1			16	2	NULL	NULL	0	NULL
110	d_assets	standard	sysRam	Mémoire RAM (Go)	number	NULL	NULL	0	0	0	1			17	1	NULL	NULL	0	NULL
111	d_assets	standard	sysDiagonal	Diagonale écran (Pouces)	number	NULL	NULL	0	0	0	1			17	2	NULL	NULL	0	NULL
112	d_assets	standard	sysOs	Système d 'exploitation	select	NULL	NULL	0	0	0	1	Données d...		18	1	7	NULL	0	width: 95%;
113	d_assets	standard	sysLicense	Licence	text	NULL	NULL	0	0	0	1			18	2	NULL	NULL	0	NULL
114	d_assets	standard	sysKey	Clef de licence	text	NULL	NULL	0	0	0	1			19	1	NULL	NULL	0	NULL
115	d_assets	standard	sysEsc	N° ESC	text	NULL	NULL	0	0	0	1	Divers		19	2	NULL	NULL	0	NULL
116	d_assets	standard	sysStoptr...	N° StopTrack	text	NULL	NULL	0	0	0	1			20	1	NULL	NULL	0	NULL
117	d_assets	standard	sysImei	N° IMEI	text	NULL	NULL	0	0	0	1			20	2	NULL	NULL	0	NULL
118	d_assets	standard	sysMeid	N° MEID	text	NULL	NULL	0	0	0	1			21	1	NULL	NULL	0	NULL
119	d_assets	standard	sysPin	Code PIN	text	NULL	NULL	0	0	0	1			21	2	NULL	NULL	0	NULL
120	d_assets	standard	sysCode	Code cadenas	text	NULL	NULL	0	0	0	1			23	1	NULL	NULL	0	NULL
121	d_assets	standard	dateCre	Date de création	date	NULL	NULL	0	0	0	1			24	1	NULL	NULL	0	NULL
122	d_assets	standard	dateMod	Date de modification	date	NULL	NULL	0	0	0	1			24	2	NULL	NULL	0	NULL

Figure 17: Table "_grids_definers", pour la section du site "Matériel (Vue standard)"

Page résultat

Seules certaines colonnes sont affichées. Ces options peuvent varier selon les paramètres issus de la table "_grids_definers", gérant les éléments de la vue "standard".

Technologies de l'information

Connecté en tant que [Nom] | [Déconnexion](#)

Accueil | Inventaires | Support | Fournitures | Mises en prêt
Gestion | Administration | ?



Développé pour vous

intégrer

[Texte flou]

Service informatique

Inventaire matériel

Vue standard

(Groupé par ville / Matériel actif uniquement / Tri par date d'acquisition descendant)

Edition

- Double-cliquer sur un enregistrement pour modifier un champ.

Export

- Seules les colonnes affichées sont exportées.
- Utiliser le filtrage des colonnes pour n'exporter que certains enregistrements.

Page 1 of 20
25
View 1 - 25 of 483

N° d'Inventaire	N° de Série	Matériel	Marque	Modèle	Commentaire
×	×	All	×	×	×
ZUG - 5 Item(s)					
[Flou]	[Flou]	POSTE DE TRAVAIL	DELL	OPTIPLEX 9010 AIO	
[Flou]	[Flou]	POSTE DE TRAVAIL	DELL	OPTIPLEX 7440 AIO	
[Flou]	[Flou]	COMMUNICATEUR	ZYXEL	GS1100	
[Flou]	[Flou]	ROUTEUR	CISCO	877	

Figure 18: Page du site section "Inventaires", vue "Matériel (Vue standard)"

Formulaire de saisie associé au tableau

Le formulaire représenté ci-dessous montre un exemple de l'ajout d'un nouvel enregistrement.

Comme on le voit, les champs "Date de modification" et "Date de création" sont préremplis par défaut, ainsi que la case à cocher "Matériel actif".

Add Record
✕

Données de base

N° d'Inventaire

N° de Série

Matériel

Marque

Modèle

Matériel actif

Commentaire

Données d'acquisition

Fournisseur

Date d'acquisition

Coût CHF (TVA incl.)

Location

Situation géographique

Ville

Adresse

Etage

Bureau

Situation interne

Groupe

Utilisateur

Réseau

Nom d'hôte

Adresse IP 1

Adresse IP 2

Adresse MAC 1

Adresse MAC 2

Système

Nom du système

Nombre de ports

Position

Téléphone

Hardware

Processeur

Nombre de cœurs

Mémoire RAM (Go)

Diagonale écran (Pouces)

Données de licences

Système d'exploitation

Licence

Clef de licence

N° ESC

N° StopTrack

N° IMEI

N° MEID

Code PIN

Code cadenas

Date de création

Date de modification

* required

Figure 19: Formulaire d'ajout d'un enregistrement

Table des illustrations

Figure 1: Structure du site web.....	19
Figure 2: Structure des répertoires du site Web.....	20
Figure 3: Répertoires du site Web (sécurité).....	21
Figure 4: Exemple de tableau PHPGrid	22
Figure 5: Structure de la page d'index du site Web	28
Figure 6: Structure de l'authentification au site Web	29
Figure 7: Structure d'exécution d'une section du site Web	30
Figure 8: Représentation du menu.....	31
Figure 9: Descriptif des champs	31
Figure 10: Mise en surbrillance du menu en regard de la page courante, affichée	53
Figure 11: Élément du menu principal mis en surbrillance	54
Figure 12: Élément de sous-menu mis en surbrillance.....	54
Figure 13: Icônes de gestion du tableau.....	95
Figure 14: E-mail reçu suite à la modification d'un enregistrement dans la table "d_assets"	96
Figure 15: Schéma bloc "assets.standard.inc.php"	97
Figure 16: Structure de la table "d_assets"	101
Figure 17: Table "_grids_definers", pour la section du site "Matériel (Vue standard)"	102
Figure 18: Page du site section "Inventaires", vue "Matériel (Vue standard)".....	103
Figure 19: Formulaire d'ajout d'un enregistrement	104