Difficultés et résolution

La première difficulté à laquelle j’ai été confronté est la connexion en php à la base de données. Bien que déjà vu, rien ne semblait fonctionner comme d’accoutumé. Etant bloqué j’ai demandé à mon tuteur de stage un peu d’aide, et il m’a indiqué que « Codeigniter », une framework php, se connecté déjà à la base, grâce à des fichiers contenant déjà tous les identifiants de MySQL. Ces fichiers sont contenu dans le répertoire « constant », dans lequel « Codeigniter » va chercher les informations qui lui intéresse pour différentes tâches (ce répertoire, pour des raisons inconnue, avait été retiré, et le site n’affiché qu’une page blanche). Cette framework doit permettre d’alléger le travail du développeur de tâches basiques. Ainsi l’exécution des requêtes et l’exploitation des données qu’elle retourne est ainsi modifié, et possède une syntaxe particulière bien que logique et similaire aux fonctions php. J’ai donc pu continuer mon travail.

Une autre difficulté rencontrée est la création d’un tableur au format .xlsx. J’ai réussi l’exportation en format .xls, un format de fichier créé par Microsoft pour sa suite Office, plus précisément Excel. Ce format concerné les versions 95 et 2003 de Excel. Alors le format .xlsx concerne les versions 2007 et 2010. Il fallait télécharger une libraire nommé « PHPExcel » permettant ceci. Mais avec « Codeigniter », l’appel à d’autre librairie demande une syntaxe particulière (comme la connexion et l’exécution de requête SQL en PHP), et placé les fichiers de la librairie dans des répertoires précis. Mon tuteur de stage m’a indiqué que la version .xls que génère déjà est complète et répond parfaitement aux besoins formulés.

Mais il s’est avéré, plus tard, que la première méthode retenu pour l’exportation dans un fichier Excel des ventes, ne soit pas une solution convenable. En effet, la solution s’appuyé sur l’échange http, avec l’utilisation de « header » dans le code PHP, qui lors de l’exécution du script, le navigateur télécharger le fichier au format .xls. Hors ce script, sera exécuté tous les jours à l’aide d’un « crontab » et de ce fait ne pourra utiliser un navigateur internet. Il donc fallu adopter la librairie « PHPExcel » qui permet d’enregistrer là où on le souhaite son fichier Excel. Le problème d’appel des librairies dans mon script, s’est avéré être étranger à « Codeigniter ». Mon tuteur de stage m’a montré un exemple d’un autre fichier qui fait appel à une librairie. Il s’agissait de la configuration d’ « include\_path ». Il s’agit des adresses absolues où se trouvent les fichiers à inclure, afin de bénéficier de la libraire. Après un moment de compréhension, j’ai réussi à exécuter l’exemple de PHPExcel fournit sur leur site. J’ai profité de la librairie pour adapter la largeur des colonnes du tableur au contenu de celle-ci, centrer tous les contenus des cellules, ainsi que de mettre en gras le nom des champs.

Mon tuteur de stage souhaitait « logger » les requêtes SQL lorsqu’une erreur survient pendant l’exportation de ventes. Mais avec « Codeigniter », l’écriture d’une requête se fait avec une syntaxe un peu spécifique. En effet lorsque qu’il y a un « where » dans la requête, on remplace les valeurs de comparaison par des « ? ». Lors de l’exécution de la requête, on passe en paramètre la requête et un tableau contenant toutes les valeurs de comparaison du « where » et dans l’ordre, pour que les données correspondes aux bons champs. Ceci ayant pour but de se prémunir des injections sql (technique de piratage), en écharpant les valeurs du « where ». De ce fait la requête ne peut être « logger » correctement, car nous n’avons pas une requête directement exécutable. Donc je me suis rendu dans la fonction qui exécute la requête passé en paramètre, et j’ai trouvé une variable qui contenait la requête avec les valeurs du « where » écharpé. A partir de là, j’ai créé une fonction dans un fichier où se trouve déjà une autre fonction qui log les erreurs sql, qui permet de journaliser la requête, et je fais appel à cette fonction au bon endroit dans la fonction de l’exécution de la requête. Cette fonction journalise n’importe quelle requête qui s’exécute, ce qui arrange mon tuteur de stage.

Dans un fichier de log que je mis en place, une même erreur PHP survient à chaque exécution de requête SQL. L’erreur survient au moment de l’appel de la fonction set\_magic\_quotes\_runtime(). En effet cette fonction n’est plus utilisée depuis la version 5.3 de PHP, alors que sur le serveur c’est la version 5.3.14 qui est utilisée. De plus l’appel à cette fonction est censé désactiver l'option magic\_quotes\_runtime, alors qu’avec cette version de PHP, il est en natif désactivé.