File d’attente

Mon tuteur de stage m’a confié un projet intéressant. Le fichier « cron.php » stocke des informations de la base dans 4 fichiers textes différents, et les envois en ftp sur un serveur. Le client qui a fourni un cahier des charges, a déjà rencontré des problèmes avec ce script. En effet, il arrive que certains fichiers ne sont pas envoyés pour des raisons diverse, comme une perte de connexion, autant du coté serveur FTP, que base de données. Mais il n’y a qu’une seule tentative d’envoi, donc si elle échoue, si le client s’en rend compte, est obligé de le signaler à l’infogérant, qui doit lui-même faire l’envoie, souvent par mail. De plus comme il y a 4 fichiers, il arrivent que seuls certains fichiers ne soient pas envoyés. Puis cela utilise beaucoup de ressources. Alors le projet doit permettre compacter les 4 fichiers et de créer une file d’attente pour l’envoie de fichiers, tant que ces derniers ne sont pas correctement envoyés.

Il faut d’abord apporter une modification à la base de données. En effet c’est dans la base que va être la file d’attente. Dans le cahier des charges, une ébauche des modifications de la base de données sont fournis. Après étude de celui-ci, et réflexion avec mon tuteur de stage, on convient que des modifications plus simple, utilisant moins de ressources et plus logique peut être mit en œuvre pour la modification de la base de données. J’ai alors rédigé le script de modifications de la base de données. Les modifications apportées sont :

* L’ajout de la table « exportCommande » dans laquelle va être stocké :
  + Un identifiant de la file d’attente.
  + Le nom du fichier a exporté.
  + La date de création de l’occurrence.
  + La date de modification de l’occurrence.
  + Le statut du fichier (« En attente » ou « Envoyé »).
  + Le nombre de tentative d’envoie
* L’ajout de la table « statutExport », dans laquelle va être stocké :
  + Un identifiant du statut.
  + Le libellé du statut.
* L’ajout d’un champ clé étrangère à la table « Commande ».
* Mise en place des contraintes d’intégrités fonctionnelles entre :
  + « exportVente » et « statutExport »
  + « commande » et « exportVente »
* Et l’ajout des 2 statuts d’exportation dans la table « statutExport ».

J’ai ensuite codé une fonction permettant de créer une archive zip, pour compresser les 4 fichiers de l’exportation. J’ai entamé des recherches, et m’es inspirant d’exemple de code trouvé ici et là, je constitue rapidement cette fonction. Des fonctions natives de php permet de créer une archive .zip. Il suffit de créer une instance de classe « ZipArchive », et utilisé la fonction « open() » de cette classe, avec comme paramètres, le nom du fichier et le mode d’utilisation de l’archive (ZipArchive::CREATE) indiquant qu’il faut créer l’archive. Puis l’utilisation du de la fonction « addFile() », pour ajouter des fichiers. Ses paramètre sont le chemin absolu du fichier et un seconde paramètre facultatif permet d’indiquer le nom du fichier à inscrire dans l’archive. Sans cela, c’est le premier paramètre qui sert pour le nom du fichier, ainsi dans l’archive se créer tous les répertoires successifs ou se trouve les fichiers, car c’est le chemin absolu. Ainsi en indiquant le nom du fichier, il se trouve à la « racine » de l’archive, sans la création de répertoire.

Je me suis ensuite attelé à la fonction ajoutant l’archive dans la file d’attente. Cette fonction doit ajouter une occurrence dans la base de données, dans la table (exportCommande) créée spécialement. Elle met également à jour les commandes concernées par l’export. En effet, grâce à la contrainte d’intégrité, chaque commande se trouve dans un et unique archive prêt à l’exportation.

Ensuite, suite à la demande de mon tuteur de stage j’ai créé un second fichier. Dans celui-ci j’ai créé une fonction qui extrait de la base de données, tous les exportations qui sont en attente d’envoi. Fonction rapidement créée, qui ne fait qu’exécuter une requête, et qui boucle autant de fois qu’il y a de fichier à envoyer.

J’ai donc entamé la création d’une fonction qui envoie les archives en SFTP. Pour cela, je me suis appuyé sur un code déjà existant dans le script « cron.php », qui envoyé les 4 fichiers textes. J’ai retravaillé ce code en envoyant qu’un seul fichier, en supprimant les instructions superflues, et en mettant en place des journalisations d’erreurs et des alertes mails.

Malheureusement, je puis réaliser des tests sur le serveur FTP du client, car mon tuteur de stage, et je le comprends, ne souhaite pas que je réalise ces tests sur le serveur du client, afin de ne pas polluer le serveur. Alors il m’a donné l’accès à l’un de serveur FTP, afin que je puis faire mes tests. COMPLETER ICI !!!

L’ajout d’un champ dans la table « commande » génère une erreur sur le passage d’une commande sur le site lui-même. En effet, lors de la commande, une requête est exécutée pour insérer une commande dans la table. Mais la requête ne spécifie pas les champs sur lesquelles de nouvelle données vont être insérer. Ainsi une erreur sql 1136 survient, car il y a moins de valeurs passées que de champs dans la table. J’ai alors modifié cette requête dans le script « Endered.php » en spécifiant les champs concernés, c’est-à-dire tout ceux présent sauf celui que j’ai moi-même rajouté. Une erreur de syntaxe d’un des champs, créer une erreur surprenante, car le style de la page ne s’affiché plus. La correction de cette erreur rétablit le comportement du site à la normal.