Note de service

Vous trouverez ci-joint la note de service concernant la mise en place d'un site web sur une machine virtuelle :

Tout d'abord, voici les composants à installer sur la machine virtuelle :

- Apache
- MariaDB
- Interpréteur PHP
- PHPMyAdmin
- Serveur SSH
- Serveur FTP
- Configurer HTTPS avec les réseaux pour sécuriser les accès sur le serveur Apache (certificat auto-signé)

Ensuite, voici les différentes actions à réaliser pour le déploiement en amont pour les SLAM :

- Exporter la base de données ***rembours_frais*** le jour j du projet commun sans les inserts de test et les utilisateurs de tests. Garder seulement les inserts de table de référence (exemple la table ville)
- Ajouter la création de la BD rembours_frais dans le script exporté de la base de données ***rembours_frais.sql***
- Créer un script de la création du compte utilisé par l'application et de ses droits sur la base de données. Ce script sera à modifier pour la création du compte en mettant l'ip de la machine virtuelle. Puis un script avec la création de d'un nouvel utilisateur délégué nommé Agnès Kintzler (sur l'application) et leurs droits dans la base de données

Egalement, voici ce que nous allons effectuer le jour j sur la machine virtuelle:

- Via FTP copier le script ***rembours_frais.sql*** sur le serveur
- Via FTP copier le dossier de l'application ***AppliRembFrais*** sur le serveur
- Importer le script SQL ***rembours_frais.sql*** dans le SGBD MariaDB. Puis exécuter le script sql ***rembours_frais.sql***
- Importer le dosssier ***AppliRembFrais*** dans un dossier prévu à cet effet
- Modification du fichier ***database.php*** dans le dossier config du dossier de l'application ***AppliRembFrais.*** Dans ce fichier, il faudra changer les variables host et port
- Configurer HTTPS avec un certificat auto-signé

Pour finir, voici ce qu'il devra être fait pour une exécution de tâches quotidiennes à réaliser sur la base de données :

- Sauvegarder la base de données tous les jours à partir de minuit

- Effectuer un job qui supprimera les infos tous les 6 mois en règle avec la loi de la RGPD

Que faut-il faire dans le code pour la table logEvenement ?

Model :

- Créer un model entity avec une classe logEvenement.php, action.php et table.php
- Créer un repository logEvenementRepository
- Créer un nouveau Repository LogEvenementRepository

Controller :

- Modifier les Controller en appelant le repository
- Ajouter une méthode avec un switch dans le cas d'un insert, select, update, delete

View :

- Créer la view ConsultLog.php cette vu affichera les logs suivant les paramètres choisi en amont par l'utilisateur (Pas indispensable) (en ajax Ramzi)

Create user '**appli_rbt_frais**'@'**adresseipdelamachine**' identified by **'JEub@e2021PgLf';**

GRANT SELECT, INSERT, UPDATE, DELETE ON rembours_frais
TO '**appli_rbt_frais**'@'**adresseipdelamachine**'

Correction :

1) **BD –>**création d'une table 'Action'
   –>création d'une table 'Table'
   –>création d'une table 'LogEvenement'

2) **MVC Application**
   Modèle :
   - **Entity** → 3 classes Action.php(id,libelle); Table.php(id,nom); LogEvenement.php(id,ipUtilisateur,dateHeure,numeroEnregistrement, Utilisateur unUtilisateur référence à la classe Utilisateur, Action UneAction référence à la classe Action et Table UneTable référence à la classe Table)

   - **Repository** →3 Repository (1 repository pour chaque classe)
     - LogEvenementRepository va contenir une méthode d'ajout (insert) avec un objet LogEvenement en entrée (ipUtilisateur,dateHeure,numeroEnregistrement, objet Action avec un id uniquement et l'objet Table avec un id uniquement)

- ActionRepository un select qui permet d'obtenir l'id à partir du libellé passé
- TableRepository un select qui permet d'obtenir l'id à partir du libellé passé

Modifier tous nos repository avec méthodes inserts dans le repository pour récupérer le dernier id de l'ajout

Contrôleur :

- 1 classe logEvenement (pas maintenant)
- Modifier tous les contrôleurs en passant le repository de logEvenement

Vue pas besoin :