AP-5		TD n°4
J2EE		29 novembre 2013

Objectifs

Accès aux bases de données

Création de classes techniques d'accès aux données
Séparation des couches MVC

Gestion des utilisateurs et des groupes (suite)

Contexte

Il s'agit du même contexte que dans le TD précédent, il est question cette fois de s'intéresser à la persistance des données : utilisateurs et groupes.
Les données relatives à ces objets seront stockées dans une base de données Mysql.

Contraintes fonctionnelles

L'application Web doit charger à son démarrage les utilisateurs, puis les groupes, en répartissant les utilisateurs dans leurs groupes respectifs.
A la fermeture de l'application, ou à la demande, les objets (utilisateurs et groupes) sont mis à jour dans la base de données (ajout, modification, suppression).

Contraintes techniques

La base de données sera stockée sur un serveur mysql au format innodb.
Lors de la modification d'un objet (utilisateur ou groupe), celui-ci sera tagué par son membre recordStatus, permettant ensuite de déterminer quelle mise à jour devra être effectuée sur l'objet au moment de la sauvegarde dans la base.
Il sera nécessaire d'ajouter un membre **recordStatus** sur la classe Bo, de type **RecordStatus**.
Le type RecordStatus sera définit en tant qu'énumération et proposera les valeurs suivantes :

- **rsNone** : par défaut, aucune modification
- **rsUpdate** : objet modifié
- **rsNew** : objet nouveau
- **rsDelete** : objet supprimé

Travail à faire

En utilisant les documents annexés :

- Créer la base de données nommée **gUsers**.
- A partir du projet **gestUserGroup-TD4** importé en fichier war, implémenter les méthodes nécessaires :
    1. au chargement des données de la base (Utilisateurs et groupes) :
        - Au lancement de l'application Web (créer un listener-lifecycle- sur le ServletContext):
            - Instancier un objet net.bo.Application
            - Avec la classe DbGateway :
                - Charger les utilisateurs
                - Charger les groupes
                - Affecter les utilisateurs aux groupes (allocateUsers)

- Stocker l'instance de **net.bo.Application** dans un attribut de nom **"app"** du servletContext
- Modifier la méthode **getSessionApp** de la classe **Utils** pour qu'elle retourne l'instance **"app"** stockée dans le servletContext.
- Sur demande, en appelant le contrôleur **reloadData** (à créer)

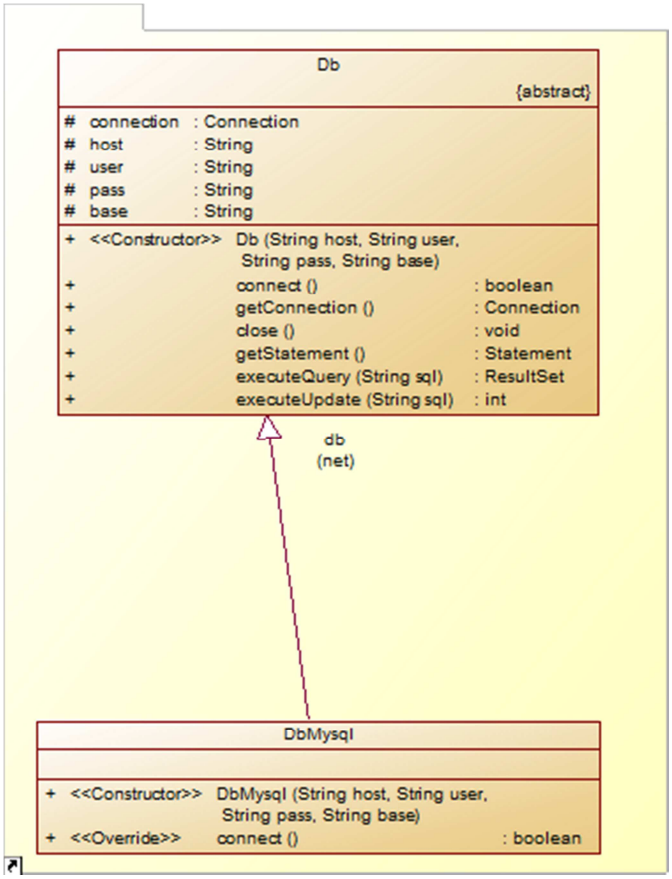
2. à la sauvegarde des données :
   - A la fermeture de l'application Web
   - Sur demande, en appelant le contrôleur **saveData** (à créer)
3. Utiliser les classes de verrouillage des enregistrements (voir net.technics dans la javadoc) pour mettre en place un verrou sur l'édition d'enregistrements par un utilisateur, de façon à éviter les modifications concurrentes par un autre utilisateur.

Le verrou doit s'activer à l'ouverture d'un formulaire de modification, et se désactiver soit à expiration (ttl passé) soit à la fermeture du formulaire (validation ou annulation)

Annexes

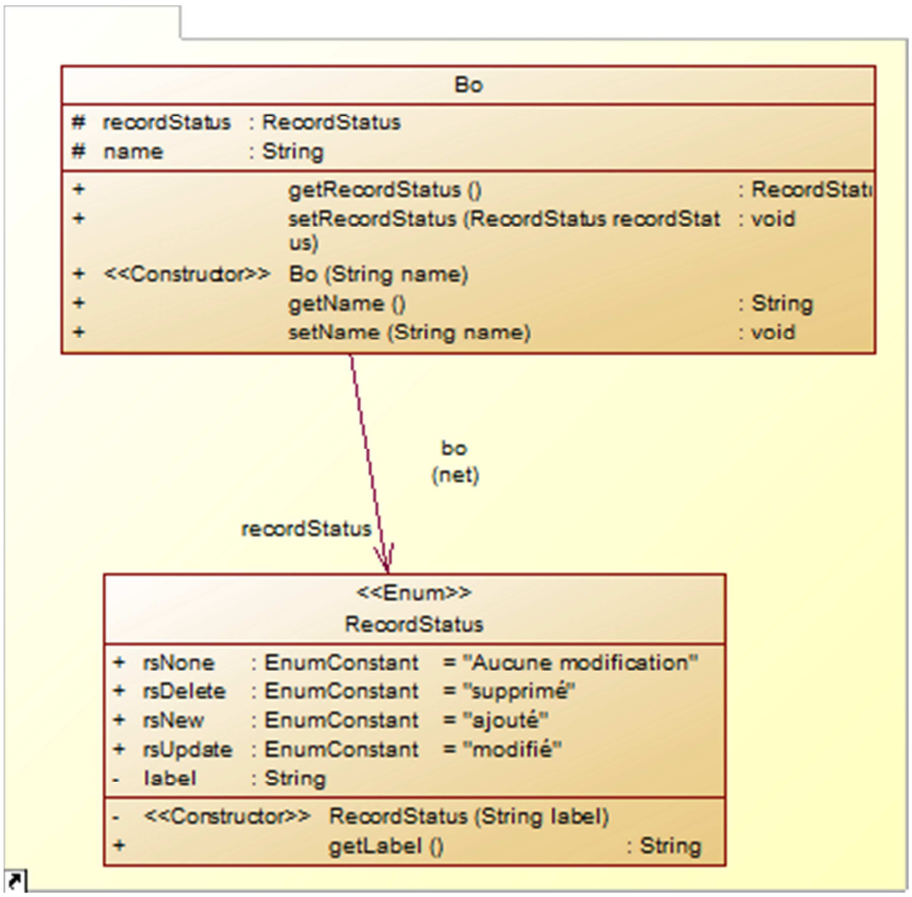
Consulter le document Accès aux bases de données en java

Classes d'accès aux données



Db.java

Mysql.java



RecordStatus.java

Classe Passerelle DbGateway.java