Liste des Web Services OBLIGATOIRES

Voila la liste des services OBLIGATOIRES que vous aurez à coder dans votre projet. Cette liste est une liste minimale et d'autres services pourront être ajoutés/spécifiés par vous-mêmes.

```
 Login
                    Entree : login + password
                    Sortie : {id,login,key}

 Logout
                    Entree : key
                    Sortie : {}


 CreateUser
                    Entree : prenom + nom + login + password
                    Sortie : {}

AddComment
                    Entree : key + text
                    Sortie : {}

AddFriend
                    Entree : key + id_friend
                    Sortie : {}


RemoveFriend
                    Entree : key + id_friend
                    Sortie : {}

Search
                    Entree : key + query + friends
                    Sortie : {Décrit lors du TD 4}
```

TD 3 : Implémentation des Web Services MySQL

Dans ce TD, nous allons nous intéresser à terminer les services de **création d'utilisateurs** et de **login**.

Rappel de Cours

Question 1. Rappelez les intérets d'utilisation d'une base de donnée relationnelle. **Question 2.** Donnez un exemple de requête SQL. **Question 3.** Quel est l'intéret de JDBC ? **Question 4.** Qu'est ce qu'un curseur ?

Tables SQL

Question 5. Donnez un exemple de programme JAVA pour traiter et afficher les résultats de votre requête en JAVA

On s'intéresse ici à définir les tables nécessairesp our notre site Web.

Question 6. Comment sera faite la création des tables ? En JAVA, Servlet, ... ? **Question 7.** Quelles sont les informations que nous souhaitons stocker ? Quelles seront les informations stockées en MySQL ? **Question 8.** Donnez une définition des différentes tables à créer **Question 9.** Comment insérer des valeurs pour tester nos services ?

MySQL et JAVA/TOMCAT

Question 10. Ecrire la fonction **DBStatic.getConnection** permettant de renvoyer une nouvelle connection à une BD Pour des raisons de concurrences, en TOMCAT, les connections ne sont pas crées manuellement, mais à l'aide d'un **DataSource**. Le Datasource est spécifié ainsi dans un fichier **context.xml** stocké dans le répertoire **META-INF**

```
<?xml version="1.0" encoding="UTF-8"?>

<Context>
    <Resource
        name="jdbc/db" type="javax.sql.DataSource"
        maxActive="100" maxIdle="30" maxWait="10000"
        url="jdbc:mysql://[host]/[database]"
        driverClassName="com.mysql.jdbc.Driver"
        username="[username]" password="[password]"
    />
</Context>
```

La création s'effectue alors à l'aide d'un objet **Database**

```
package tools;

import java.sql.Connection;
import java.sql.SQLException;

import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;

public class Database {
```

```
    private DataSource dataSource;

    public Database(String jndiname) throws SQLException {
       try {
          dataSource = (DataSource) new InitialContext().lookup("java:comp/env/" +
              jndiname);
       } catch (NamingException e) {
          // Handle error that it's not configured in JNDI.
          throw new SQLException(jndiname + " is missing in JNDI! : "+e.getMessage());
       }
    }

    public Connection getConnection() throws SQLException {
       return dataSource.getConnection();
    }


}
```

et la création de connection s'effectue ainsi:

```
    public static Connection getMySQLConnection() throws SQLException
    {
          if (DBStatic.mysql_pooling==false)
          {
                return( DriverManager.getConnection("jdbc:mysql://"+DBStatic.mysql_host
                    +"/"
                +DBStatic.mysql_db, DBStatic.mysql_username,DBStatic.mysql_password));
          }
          else
          {
                if (database==null)
                {
                      database=new Database("jdbc/db");
                }
                return(database.getConnection());
          }
    }
```

3.1 Service de création d'un utilisateur

On rappelle l'organisation du service de création d'un utilisateur
Fonction: `JSONObject createUser(String login, String password, String nom, String prenom)`

1. Verification des parametres (il ne faut pas que les paramètres valent null)
2. Verifier si un utilisateur avec le même login existe déjà. Si oui => erreur 1
3. Ajouter l'utilisateur à la base de données
4. Renvoyer OK

Question 1. Ecrire la fonction permettant de vérifier si l'utilisateur existe déjà dans la base de données **Question 2.** Ecrire la fonction permettant d'insérer un utilsiateur dans la base de données

3.2 Service de login

Pour rappel, le service de login s'organise ainsi:

```
if ((login==null) || (password==null))
                return(ServicesTools.error("Wrong Arguments",0));

            try
            {
                //Verifie que l'utilisateur existe sinon ERROR 1
                boolean is_user=AuthentificationTools.userExists(login);
                if (!is_user) return(ServicesTools.error("Unknown user "+login,1));

                //Verifie que le password et l'utilisateur sont OK sinon ERROR 2
                boolean password_ok=AuthentificationTools.checkPassword(login,password)
                    ;
                if (!password_ok) return(ServicesTools.error("Bad password "+login,2));

                //Récupère l'id de l'utilisateur
                int id_user=AuthentificationTools.getIdUser(login);

                JSONObject retour=new JSONObject();
                //Insère une nouvelle session dans la base de données
                String key=AuthentificationTools.insertSession(id_user,false);
                retour.put("key",key);

                return(retour);
            }
            catch(JSONException e)
            {
                return(ServicesTools.error("JSON Problem "+e.getMessage(),100));
            }
            catch (BDException e)
            {
                return(ServicesTools.error("Problem while generating session key",1000)
                    );

            } catch (Exception e) {
                return(ServicesTools.error("Problem...",10000));
            }
```

Question 1. Ecrire les fonctions manquantes