Interface à la base de données

L'interface à la base de données se compose de trois parties : une partie serveur et deux partie client, l'une en java pour Android, l'autre en javaScript pour la page web.

La partie serveur se fait via des scripts PHP. Il existe deux scripts PHP, le premier pour écrire dans les tables de la BD est le plus souvent adressée par la page web, le second script sert à la lecture et est, pour l'heure, principale utilisé par le client Android. Toutes les requêtes au script PHP se reçoivent une structure JSON en réponse.

Les partie clients Android se fait via une série de classes DbAccess. Ces classes supportent 2 type de bases de données : une locale et une distante. Le module d'accès distant envois les requêtes aux scripts PHP et décodent les réponses retournées pour encapsuler, s'il y a lieu, le résultat dans des classes facile à manipuler par l'interface usager. Le module d'accès local, gère, sous SQLite, une image qui est un sous ensemble de la base de donnés distante.

# PHP

Toutes les requêtes aux scripts PHP reçoivent une réponse en format JSON du type :

{«Status» : «Success/Fail», [data]}

Il existe 4 scripts PHP :

|  |  |  |
| --- | --- | --- |
| index.pĥp | - | Sert principalement à permettre au client de valider la présence du serveur, prévient une forme superficielle de hackage. |
| connexion.php | - | Établi la connexion avec la base de donnée. |
| Writer.php | - | Script permettant d'écrire dans la base de données. |
| Reader.php | - | Script de lecture de la base de données. |

Les deux dernier fichier sont ceux actuellement adressé par les diverse applications client. Ce sont eu qui formatent les requêtes à la base de données selon les paramètres passé par le client.

Le script «Writer.php» est le plus simple. Les routines ont une correspondance 1 pour 1 avec les tables de la base de données. Ces routines sont :

|  |  |  |  |
| --- | --- | --- | --- |
| newLigue |  | Permet d'inscrire une nouvelle ligue. Ses paramètres sont : | |
|  | NomLigue - le nom de la ligue  idGestionnaire – Id du fondateur de la ligue |
| newPersonne |  | Permet d'inscrire une nouvelle fiche Personne. Ses paramètres sont : | |
|  | Nom  prenom  dateDeNaissance (facultatif)  telephone (facultatif)  courriel (facultatif)  adresse (facultatif) |
| newEquipe |  | Enregistre une nouvelle équipe, liée à un ligue et une sous-ligue | |
|  | NomEquipe – nom de l'équipe  idLigue – id de la ligue auquel l'équipe appartient  idSousLigue – id de la sous ligue auquel l'équipe appartient. (facultatif) |
| newAlignement |  | Inscrit une nouvelle entrée dans la table des Alignements | |
|  | IdJoueur – id du joueur  idEquipe – id de l'équipe  idSaison – id de la saison  position – Position joué  numeroChandail – Numéro du chandail  temporaire – marque si le l'alignement est temporaire (V/F) |
| newSaison |  | Enregistre une nouvelle saison | |
|  | IdLigue – id de la ligue  dateDebut – Date de début de la saison  dateFin – Date de fin de la saison  numeroSaison – Numéro de la saison |
| newPartie |  | Inscrit une nouvelle Partie dans la table. | |
|  | Lieu – Lieu ou la partie  duree – Durée de la partie (facultatif)  date – Date où la partie a lieu  idEquipe1 – Id de l'équipe 1  idEquipe2 – Id de l'équipe 2  pointage1 – Pointage final équipe1 (facultatif)  pointage2 – Pointage final équipe2 (facultatif) |

Le script «Reader.php» plus complexe car il doit combiner les information de plusieurs tables. Ces routines sont :

validateLogin – Valide le nom d'usager et le mot de passe.

Paramètres :

userName – nom d'usager

passsWord – mot de passe e l'usager

retourne : un objet contenant

- le status (Success/Fail)

- un objet contenant l'id, nom et prénom de l'usager

- Un objet contenant une liste de Compétences

listeEquipe – fait la liste des équipes d'une ligue

Paramètres :

idLigue – Id de la ligue

retourne : un objet contenant

- le status (Success/Fail)

- Un objet contenant une liste d'équipes

listeJoueur – Fait la liste des joueurs d'une équipe

Paramètres :

idEquipe – Id de l'équipe

retourne : un objet contenant

- le status (Success/Fail)

- Un objet contenant une liste de joueur

listeJoueurLigue – Fait la liste des joueurs d'une ligue

Paramètres :

idLigue – Id de la ligue

retourne : un objet contenant

- le status (Success/Fail)

- Un objet contenant une liste de joueur

listeLigue – Fait la liste des ligues d'un gestionnaire

Paramètres :

idGestionnaire – Id du gestionnaire

retourne : un objet contenant

- le status (Success/Fail)

- Un objet contenant une liste de ligues

listeLigueParMarqueur – Fait la liste des ligues pour lesquels un marqueur est autorisé

Paramètres :

idMarqueur – Id du marqueur

retourne : un objet contenant

- le status (Success/Fail)

- Un objet contenant une liste de ligues

listeGestionnaires

Paramètres :

aucun

retourne : un objet contenant

- le status (Success/Fail)

- Un objet contenant une liste de gestionnaires

# Android