Pense bête pour l'execution et la récupération de données

Mithrandir

Pense bête requête SQL

Bonjour,

La récupération de données est vachement utile pour les scripts CPP, c'est pourquoi je vous partage ce petit pense bête :)

Comme vous le savez, AzerothCore contient 3 base de données : `auth` `characters` `world`

Pour chaque base de données une définition existe ainsi :

```
//auth
LoginDatabase.une_methode
//Characters
CharacterDatabase.une_methode
//world
WorldDatabase.une_methode
```

Maintenant que l'on connaît ces définitions il faut utiliser deux méthodes : `PQuery` ou `PExecute`

- `PQuery` Execute et renvoie des données de type QueryResult (`SELECT`)
- `PExecute` execute simplement une requête (`UPDATE` , `DELETE` , `INSERT`) et ne renvoie rien : type `void`

```
QueryResult query = LoginDatabase.PQuery("SELECT * FROM table WHERE col = %u", 52)
LoginDatabase.PExecute("INSERT INTO ...") /*void*/
```

Mais c'est quoi le `%u` ?

Le `%u` est un caractère qui sera remplacé par les arguments mis après le string, MAIS attention ils veulent tous dire quelque chose !

- `%u` pour uint (entier non signé (0 à un nombre))
- `%s` pour un string
- `%i` pour un int (entier signé (-un nombre à +un nombre)
- etc

Revenons à notre PQuery

Donc comme je le disais `PQuery` renvoie un résultat de type `QueryResult` qui si on regarde bien le code n'est qu'un `typedef` d'un `std::shared_ptr<ResultSet>` donc `QueryResult` est un pointeur intelligent de ResultSet, donc avant de faire quelque chose, regarder si le résultat est différent de `nullptr` ! :)

Vous allez pouvoir maintenant récuperer les résultats de votre requête :) voici ma méthode :

```
//Ce n'est qu'un exemple :)
QueryResult result = LoginDatabase.PQuery("SELECT username, id FROM account");

if(result) {
    do {

        Field* fields = result->Fetch();
```

```
        std::string username = fields[0].GetString();
        uint32 id = fields[1].GetUInt32();

    } while(result->NextRow());
}
```

Pour `Fetch` le résultat on utilise un pointeur sur Field qui contiendra toutes les informations d'un résultat, la condition dans le while nous permet de savoir si l'on a encore des données en réserve :)

J'espère que ce petit Pense-Bête vous sera utile :)

---

iThorgrim

Merci beaucoup pour ce tutoriel :D

---

noc

Que voila une bonne base pour creer plein de choses, merci
