

Une erreur est survenue lors du chargement de la version complète de ce site. Veuillez vider le cache de votre navigateur et rafraîchir cette page pour corriger cette erreur.

Stocker des informations avec Eluna

iThorgrim

Stocker des informations avec Eluna

Tutoriel par @iThorgrim-hub

Bonjour, aujourd'hui un tutoriel plutôt complexe dans la forme plus que dans le fond ..Comme vous le savez pour les serveurs World of Warcraft il existe 2 langages de développement qui sont les vedettes et les plus utilisés le SQL et le C++.

Depuis maintenant quelques années un module à fait sont entrés et à permis à des petits devs de faire de grande choses, je parle de ELUNA, qui permet de développer en LUA pour les émulateurs compatibles.

Alors Eluna est plein de bonne choses, mais disons que pour ce qui est du stockage c'est presque peine perdu.

Quand vous voulez stocker une information, vous êtes obligés, de faire une requête SQL et pour pas vous mentir, si vous avez un script de Modificateur de rates EXP et que vous souhaitez vérifier la rate choisis à chaque Kill, ça fait beaucoup de requêtes pour pas grand choses .. J'étais dans ce problème, et j'ai trouvés une solution, j'ai utilisés des Tableaux pour stocker des informations(C'est pas la meilleur mais c'est celle que je préfère).

Et aujourd'hui je vous propose un guide plus qu'un tutoriel, car les domaines d'applications sont vraiment vastes et que j'ai pas d'exemple ultra stylés pour vous montrer le **p p p power** de cette chose.

Pour notre exemple on va dire que l'ont cherche à stocker les informations de notre personnage et plus précisément la Rate d'EXP.Donc dans l'exemple on a fait notre code, la rates d'exp elle est stockés dans une colonne custom de characters qui s'appelleras rates. (je rappelle que ceci est un exemple, ne cherchais pas la colonne rate dans characters hein ..)

Alors ont va juste improviser un bout de code pour avoir un petit exemple « visuel » sous les yeux :

```
local EXP = {};  
local PLAYER_VENT_ON_LOGIN = 3;  
function EXP.GetRates(event, player)  
end  
RegisterPlayerEvent(PLAYER_EVENT_ON_LOGIN, EXP.GetRates)
```

Donc pour faire assez simple et concis on utilise l'événement connexion pour aller chercher la Rates d'EXP et la garder en mémoire le temps que le joueur reste connectés!Maintenant on va se créer un tableau et des valeurs par défauts (au cas ou..).

```
local EXP = {};  
local PLAYER_EVENT_ON_LOGIN = 3;  
  
function EXP.GetRates(event, player)  
    local pGuid = player:GetGUIDLow()  
    if not(EXP[pGuid])then  
        EXP[pGuid] = {  
            rate = 1  
        }  
    end  
end  
RegisterPlayerEvent(PLAYER_EVENT_ON_LOGIN, EXP.GetRates)
```

Alors là je vais vous décortiquer ça :

- **local pGuid = player:GetGUIDLow()** : On va chercher le GUID du joueur qui je le rappelle est unique ! (Très important d'avoir une

donnée unique comme Index/Identifiant)

- **if not(EXP[pGuid])then** : Si le Tableau EXP[pGUID] n'existe pas, donc si le tableau avec notre GUID n'existe pas alors ..
- **EXP[pGuid] = { rate = 1}** : Tu me créer un tableau avec dedans une "colonne" rate qui a par défaut la valeur 1

Attention je vulgarise la chose, en réalités certains terme ne s'utilise/se dise pas comme ça, c'est pour l'exemple et le guide.

Alors du coup maintenant on a notre valeur par défaut et notre tableau il nous faut aller récupérer la rate de notre joueur.

```
local EXP = {};  
local PLAYER_EVENT_ON_LOGIN = 3;  
  
function EXP.GetRates(event, player)  
    local pGuid = player:GetGUIDLow()  
    if not(EXP[pGuid])then  
        EXP[pGuid] = {  
            rate = 1  
        }  
    end  
  
    local Q0 = CharDBQuery('SELECT rate FROM `characters` WHERE guid = '..pGuid..');  
    if Q0 then  
        EXP[pGuid].rate = Q0:GetUInt32(0)  
    end  
end  
RegisterPlayerEvent(PLAYER_EVENT_ON_LOGIN, EXP.GetRates)
```

Comme pour tout à l'heure je vais expliquer un peu :

- **local Q0 = CharDBQuery('SELECT rate FROM characters WHERE guid = '..pGuid..');** : La très simple je vais chercher la rate de mon personnage
- **if Q0 then** : Si Q0 aboutis à quelques choses (car il ne peu pas être nul si jamais la rate devrais normalement être à 1)
- **EXP[pGuid].rate = Q0:GetUInt32(0)** : Alors rate dans EXP[pGuid] est égale au résultat de la requête (Donc si le résultat est 28, EXP[pGuid].rate seras maintenant égale à 28)

Voilà, je ne sais pas si vous avez compris ou si c'est assez clair, mais ça permet beauuucoup plus de choses et ça rajoute un confort appelés stabilités des serveurs. Si jamais voici quelques petits conseil que je peu vous donner pour l'utilisation des tableaux 'Stockeur' :

****Arrêtez le stockage en live dans la base de donnée et préférés des stockage directement dans des tableaux, puis stockés seulement dans la DB à la déconnexion du joueur.**

Arrêtez les vérifications systématique dans la base de donnée et stockés l'information une fois dans un tableau J'espère que ceci vous aideras à stocker des informations, et vous pouvez littéralement en abuser.

Arrête avec tes 300 requêtes/scd. Allez moi je go manger des bébé phoques..**