

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.

Parser Python : Drop Gold de tous les PNJs depuis wowhead

Galathil

Important : Ce tutoriel a été rédigé par Nobody pour le site wow-emu.fr. Je le recopie ici à l'identique, Il se peut que certaines informations soient obsolètes à ce jour. Bonne lecture !

Parser Python : Drop Gold de tous les PNJs depuis wowhead

Bonjour à tous,

voici aujourd'hui un petit script réalisé en python qui a pour objet de générer une requête SQL permettant de mettre à jour tous les drops de Gold des PNJs d'une zone, à partir des données de wowhead.

Le script fonctionne de façon assez simple :

1. On récupère la liste des pnjs se situant dans une zone X depuis wowhead (la fonction est appelée pour un certain nombre de zones à la fin du script)
2. Pour chaque pnj de cette liste, on récupère sa page wowhead correspondante puis on parse le XML pour récupérer les valeurs qui nous intéressent (ici, le drop de Gold)
3. On génère une requête SQL UPDATE à partir de ces données pour les intégrer dans la table creature_template

L'objectif n'est pas tellement d'utiliser ce script comme tel, mais de s'en servir comme base modifiable à souhait pour trouver d'autres données. Par exemple, générer des requêtes pour traduire les quêtes, récupérer les loots des PNJs, identifier l'ensemble des objets vendus par des PNJs, etc etc. Vous vous doutez bien que la liste est infinie ☐

~~Voilà le script donné tel quel, sans commentaires. A savoir qu'un tuto reprenant en partie ce script est en cours de rédaction. Celui-ci ne reprendra pas la totalité du script proposé ici afin de garder une certaine exclusivité pour les membres confirmés ☐~~

```
import os, re
from urllib import request

def writeSQLQuery(file, entry, copper, silver, gold) :
    money = gold * 10000 + silver * 100 + copper
    minMoney = 0
    maxMoney = 0
    if (gold != 0 and silver != 0 and copper != 0) :
        minMoney = money - 50000
        maxMoney = money + 50000
    elif (gold != 0 and silver == 0 and copper == 0) or (gold == 0 and silver != 0) :
        minMoney = money - 500
        maxMoney = money + 500
    elif (gold == 0 and silver == 0) :
        minMoney = money - 5
        maxMoney = money + 5

    sqlQuery = "UPDATE creature_template SET mingold = " + str(minMoney) + ", maxgold = " + str(maxMoney)
    sqlQuery = sqlQuery + " WHERE entry = " + str(entry) + ";"
    file.write(sqlQuery)

def parseMoney(entries, zoneId):
    fileSQL = open("E:/money_" + str(zoneId), 'w')
```

```

print("OK")
for element in entries :
    url = "http://www.wowhead.com/npc=" + str(element)
    copper = 0
    silver = 0
    gold = 0

    u = ""
    try :
        u = request.urlopen(str(url))
    except :
        print("Unable to parse entry " + str(element))
        continue

    #print("Parsing" + str(element))
    content = str(u.read())
    moneyTotal = 0
    globalMoneyRe = re.search(r'money\\x3D([0-9]*)', content)
    #print(globalMoneyRe)
    if globalMoneyRe :
        moneyTotal = int(globalMoneyRe.group(1))

    gold = int(moneyTotal / 10000)
    moneyTotal -= 10000 * gold

    silver = int(moneyTotal / 100)
    moneyTotal -= 100 * silver

    copper = int(moneyTotal)
    moneyTotal -= copper

    #print(str(gold) + ' ' + str(silver) + ' ' + str(copper))

    if(gold != 0 or silver != 0 or copper != 0):
        writeSQLQuery(fileSQL, element, copper, silver, gold)

fileSQL.close()

def parseEntriesInZone(zoneId) :
    url = "http://www.wowhead.com/zone=" + str(zoneId)
    u = ""
    try :
        u = request.urlopen(str(url))
    except :
        return 0

    print('OK')
    content = u.read()
    content = str(content)

    fileSQL = open("E:/Content.txt", 'w')
    #print(str(content))
    liste = list()
    globalRe = re.search(r"template: 'npc', id: 'npcs'.*?}.?};", content)
    if globalRe :
        print("OK")
        globAsString = str(globalRe.group(0))
        fileSQL.write(globAsString)
        eachIdRe = re.search(r'"id":([0-9]*)', globAsString)
        while eachIdRe :

```

```
print("Got" + str(eachIdRe.group(1)))
liste.append(eachIdRe.group(1))
globAsString = globAsString[eachIdRe.start() + len(eachIdRe.group(0)):]
eachIdRe = re.search(r'"id":([0-9]*)', globAsString)
```

```
parseMoney(liste, zoneId)
```

```
parseEntriesInZone(6661)
parseEntriesInZone(6134)
parseEntriesInZone(5841)
parseEntriesInZone(6142)
parseEntriesInZone(6141)
parseEntriesInZone(5785)
parseEntriesInZone(6138)
parseEntriesInZone(6006)
parseEntriesInZone(6757)
parseEntriesInZone(5842)
parseEntriesInZone(5840)
parseEntriesInZone(5805)
```