

2H Házi feladat

Konfigurációkezelés

FIGYELEM: A házi feladat megoldása előtt olvassuk el a tárgy weblapján lévő HF tudnivalókat! A házi feladat leadása előtt nézzük végig a HF tudnivalóknál szereplő ellenőrző listát!

Feladat

Cégünknek eddig nem centralizáltan kezelték a felhasználókat, így az évek alatt elszaporodtak az egyes gépeken található felhasználói fiókok és csoportok. Az új rendszer megtervezése és üzembe helyezése előtt szükség van a jelenlegi helyzet felmérésére. Ahelyett, hogy papírral és ceruzával bejárnánk az összes irodát, írjunk egy olyan szkriptet, amely képes a hálózatunkon lekérdezni a bekapcsolt gépek csoport és felhasználói adatait és azt megfelelően összefoglalva visszaadni.

Készítsünk tehát egy **Python 3** szkriptet, ami **CIM-XML** segítségével összegyűjti a szükséges adatokat. A fejlesztés és tesztelés során feltételezhetjük, hogy az összes távoli számítógépen **Linux** operációs rendszer fut.

A szkript elnevezése és paraméterezése

```
get_users.py [-s SOURCE] [-o OUTPUT] [-v]
```

A szkriptnek kötelező ezt az elnevezést és paraméterezést használnia. A szkript használjon nevesített paramétereket. A paraméterek sorrendje ne legyen megkötve.

A szkript a paramétereket a következő formában fogadja:

- `-s SOURCE`, `--source SOURCE`
a forrás JSON fájl elérési útvonala
- `-o OUTPUT`, `--output OUTPUT`
a kimeneti mappa útvonala
- `-v`, `--verbose`
opcionális paraméter, megadása esetén több információt adunk meg a felhasználókról

Példa a szkript lehetséges használataira:

```
python3 get_users.py -s machines.json -o ./results
python3 get_users.py -v -o /home/meres/Desktop/results --source net.json
```

Bemeneti fájl

A bemenet egy JSON fájl a mellékelt struktúrával. Megadhatunk egyszerre több konfigurációt, a szkriptnek az összes szerverről le kell tudnia kérdeznie a releváns adatokat.

```
{
  "Connections": [
    {
      "Configuration": {
        "Host": "localhost",
        "Username": "meres",
        "Password": "jelszo"
      }
    }
  ],
}
```

Intelligens rendszerfelügyelet (VIMIA370)

```
{
  "Configuration": {
    "Host": "192.168.183.145",
    "Username": "meres",
    "Password": "jelszo"
  }
}
```

A házi feladat fejlesztése és tesztelése során elég csak a http protokollra felkészülni, a https-t nem kell támogatni.

(A jelszó nyílt szövegben tárolása éles környezetben nem javasolt megoldás, ez most csak a házi feladat egyszerűsége miatt engedhető meg. Éles környezetben a jelszót érdemes ilyenkor például titkosítva tárolni vagy nyilvános kulcsú titkosításra alapuló módszereket alkalmazni.)

Kimenet

A kimenet **több JSON** fájl a megadott kimeneti mappában, melyeket a forrás fájlban lévő **host** neve azonosít, például „localhost.json”. Ebben a talált csoportokat felsoroljuk, és azokon belül tömb formájában a hozzájuk tartozó felhasználókat is. A JSON behúzás **két** szóköz hosszúságú legyen.

Alap esetben csak a csoport és user neveket írjuk ki, azonban a verbose opcionális paraméter megadásával kisorosítjuk a felhasználók azonosítóit és home könyvtárait is.

Például verbose nélkül (részlet):

```
{
  "Groups": [
    {
      "Name": "root",
      "Users": [
        "root",
        "sync"
      ]
    },
    {
      "Name": "meres",
      "Users": [
        "meres"
      ]
    }
  ]
}
```

Verbose paraméterrel:

```
{
  "Groups": [
    {
      "Name": "root",
      "Users": [
        {
          "Name": "root",
          "Home": "/root",
          "UserID": "0"
        },
        {
          "Name": "sync",
          "Home": "/sbin",
          "UserID": "1"
        }
      ]
    },
    {
      "Name": "meres",
      "Users": [
        {
          "Name": "meres",
          "Home": "/home/meres",
          "UserID": "1000"
        }
      ]
    }
  ]
}
```

Intelligens rendszerfelügyelet (VIMIA370)

További elvárások

- A szkript kezelje azt az esetet, ha a távoli géphez nem sikerül csatlakozni. Ilyenkor ne készítsünk az adott géphez tartozó kimenetet, adjunk hibaüzenetet futás közben, majd folytassuk a feldolgozást.