



TECHNICKÁ DOKUMENTACE

Název: ČTI OS

Účel: Skript posílá požadavek na službu ČTI OS na základě pole pseudonymizovaných oprávněných subjektů (POSidentů) a ukládá odpověď do SQLITE databáze Export_vse.db

Datum: červen 2019

Copyright: (C) 2019 Linda Kladivová

Email: l.kladivova@seznam.cz

Úvod

Tento skript byl zpracován v rámci předmětu 155yfsg 2019 na fakultě stavební ČVUT. Není v konečné verzi, v budoucnu z něj bude vytvořena knihovna, která bude zakomponována do pluginu VFK v softwaru QGIS. V případě dotazů, nejasností, při návrhách na vylepšení se obraťte na uvedenou emailovou adresu.

Návod ke skriptu

Pro správný chod skriptu je nutno si připravit několik souborů. Zaprvé textový soubor s pseudonymizovanými oprávněnými subjekty s názvem posidents.txt. Zadruhé soubor request.xml s přístupovým jménem a heslem do služby ČTI OS. Zatřetí stáhnout si databázi Export_vse.db. Vzorové ukázky souborů request.xml a posidents.txt jsou součástí projektu zde na GitHubu. Aby skript fungoval správně, je nutno mít tyto 3 jmenované soubory ve stejném adresáři jako skript cti_os.py. Příklad správného sestavení vstupních dat je uveden na následujícím obrázku 1.

Název	Stav	Datum změny	Typ	Velikost
cti_os.py	✓	18.06.2019 19:24	Python File	15 kB
Export_vse.db	✓	21.05.2019 15:59	Data Base File	11 620 kB
posidents.txt	✓	18.06.2019 16:49	Textový dokument	2 kB
request.xml	✓	17.06.2019 9:38	Dokument ve for...	1 kB

Obrázek 1: Struktura adresáře před vykonáním skriptu

Funkcionalita skriptu

Vytvořen se LOG soubor (funkce `create_log_file`), do kterého se budou později vypisovat chybné posidenty a statistiky o počtu správně zpracovaných posidentů a o počtu konkrétních chyb: NEPLATNY IDENTIFIKATOR, EXPIROVANY IDENTIFIKATOR, OPRAVNENY SUBJEKT NEEXISTUJE. LOG soubor se vytvoří ve stejném adresáři, jako jsou všechny výše jmenované soubory.

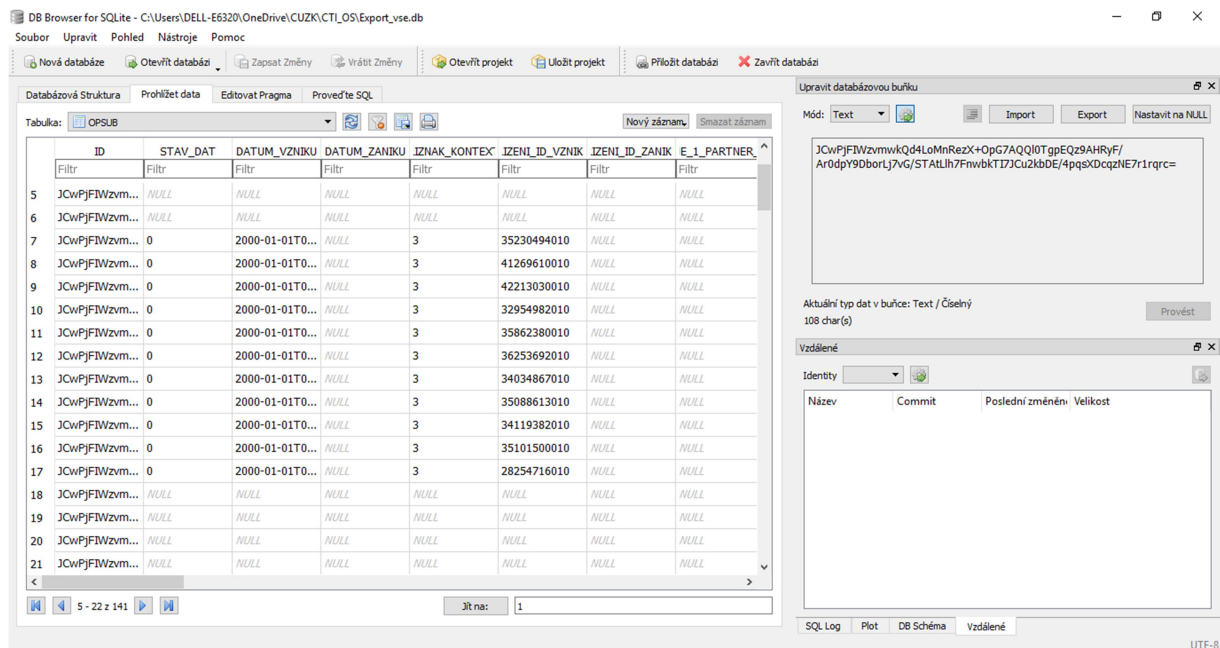
Dále se otevře se soubor s polem posidentů a zkontroluje se, že nemá žádné duplicity, pokud ano odstraní se (funkce `remove_duplicates`). Pokud bude v poli více než 100 posidentů, je dotaz na službu rozdělen do vícero dalších dotazů. Pokud je počet posidentů menší než 100, nic se nerozděluje.

Vzhledem k tomu, že jména atributu v XML dokumentu v několika případech není možné jednoduše převést na jména sloupců v databázi, bylo nutné pro tyto speciální případy definovat převodní slovník.

Podle pole posidentů se sestaví XML a to ve funkci `draw_up_xml_request` (maximálně pro 100 posidentů). Vstupem do této funkce je cesta k souboru request.xml, ve kterém se může změnit jméno a heslo, ale žádné posidenty se nevyplňují, ty jsou obsahem souboru posidents.txt. Další funkce `call_service` zavolá službu CTI OS a služba vrátí odpověď, která bude poměrně rozsáhlá - velký XML soubor, který bude obsahovat atributy k jednotlivým posidentům. Je nutné, aby HTTP status kód byl 2xx, což je ve funkci také kontrolováno.

Dále byly vytvořeny dvě funkce `transform_names` a `transform_names_dict` na konverzi jmen atributu v XML dokumentu na názvy sloupců v databázi. První funkce se zabývá jednodušším případem konverze, druhá využívá speciální převodní slovník. V další funkci `save_attributes_to_db` se postupně prochází všechny podatributy v atributu os a za pomoci funkce na konverzi jmen se vkládají postupně všechny tyto podatributy do databáze. Hned na začátku for cyklu je atribut os kontrolován, jestli neobsahuje podatribut chybaPOSIdent, který může obsahovat hesla: NEPLATNY IDENTIFIKATOR, EXPIROVANY IDENTIFIKATOR, OPRAVNENY SUBJEKT NEEXISTUJE. Pokud se chyba najde, tak se vypíše chyba (podle situace) do log souboru a zbytek for cyklu se přeskočí a jde se na další posident. V databázi se ještě vytvoří sloupeček OS_ID, do kterého se vloží podatribut osId. Takto se for cyklus vykoná postupně pro každý atribut os v XML souboru (tedy tolikrát, kolik je posidentů v zadaném seznamu).

Na následujícím obrázku 2 je databázová struktura po vykonání skriptu. Několik posidentů se aktualizovalo.



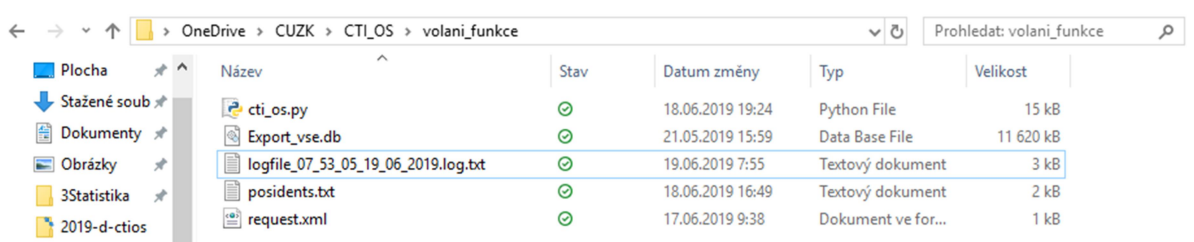
Obrázek 2: Databáze po aktualizaci v databázovém prostředí DB Browser for SQLite

Příklad log souboru je vidět na následujícím obrázku 3.



Obrázek 3: Log soubor informací a statistikami o průběhu skriptu

Struktura adresáře po vykonání skriptu je zřejmá z obrázku 4:



Obrázek 4: Struktura adresáře po vykonání skriptu