Data-flow Architecture

Pipe and Filter

За потребите на нашата апликација, ние имплементиравме Pipe and Filter преку кој податоците кои ни се потребни ќе ги добиеме, филтрираме и ќе ги внесеме во база на податоци.

Почетни информации

Скриптите се bash scripts во UNIX базирани системи.

Базата на податоци која ја искористивме е PostgreSQL Database. Доколку немате променето основни поставки при инсталација, database host id е 127.0.0.1 и database port number е 5432.

Повик на скрипта и потребни влезни податоци

За да се користи филтерот (имплементиран во скриптата pipe-and-filter.sh) потребно е да навигираме до директориумот во кој истиот се наоѓа. Во овој директориум, заедно со скриптата, треба да се и потребните документи т.е. database.sh и osmfilter-от .

При повик за извршување на скриптата pipe-and-filter.sh, потребно е да се испратат и четири параметри кои претставуваат информации за тоа во која база на податоци да се зачуваат податоците (host ip, port number и database user) и локацијата на документот со податоци (data location path). Повикот на истата во UNIX базирани системи се врши со наредбата bash ./pipe-and-filter.sh или само ./pipe-and-filter.sh по која се проследени четирите влезни параметри. При извршување, ќе биде потребно е да се внесе и лозинка на корисникот на системот и лозинка на корисникот на базата на податоци.

```
./pipe-and-filter.sh <host_ip> <port_number> <database_user> <data_location_path>
```

Имплементација на pipe-and-filter.sh

Најпрво имаме повик кон database.sh скриптата, во која со основни команди за работа со PostgreSQL база на податоци, ние се конектираме со database серверот. Потоа креираме (доколку не постоела) нова база на податоци и во неа креираме (доколку не

постоела) нова табела/релација во која понатаму ќе ги зачуваме потребните податоци за работа на апликацијата.

Доколку конекцијата со базата на податоци (и креирањето на нова база и нова табела доколку е потребно) е успешно, извршувањето на pipe-and-filter.sh скриптата ќе продолжи со осврт на osmfilter-от.

Со користење на osmfilter, од почетните податоци во map.xml, за потребите на нашата апликација, ние ги задржуваме податоците за:

- id (идентификациски број на паркингот, задолжително мора да постои),
- lat и log (координати за локација на паркингот, задолжително мора да постои),
- како и name(име),
- capacity(капацитет),
- access(пристап),
- fee(цена),
- operator(оператор, сопственик),
- website(веб-страна),
- supervised(дали е надгледувана) и
- parking(тип на паркинг - катна гаража, подземна итн.) .

Податоци ги зачувуваме во .csv датотека, по еден влез/ред за секој паркинг, и за истите внимаваме да бидат уникатни. Понатаму, секој посебен влез го зачувуваме во табела во базата на податоци која претходно ја креиравме.

Краен резултат од извршувањето на pipe-and-filter.sh

По извршувањето на скриптата, базата на податоци ќе биде целосно пополнета или дополнета со новите податоци со тоа што цевката која ја имплементиравме ни функционира и кога базата на податоци не постои и кога веќе постои.

```
greengummybear@DESKTOP-IS1VHPO:/mnt/c/Users/veron/Documents/GitHub/DIANS-P
tecture$ bash ./pipe-and-filter.sh 127.0.0.1 5432 postgres ../data/map
Reading package lists... Done
Building dependency tree
Reading state information... Done
osmctools is already the newest version (0.9-2).
0 upgraded, 0 newly installed, 0 to remove and 128 not upgraded.
Password for user postgres:
Password for user postgres:
Database creation finished...
Password for user postgres:
ERROR:  duplicate key value violates unique constraint "parking_pkey"
DETAIL:  Key (id)=(170716212) already exists.
```