Dokumentácia - iDnes scraper do grafu

4IZ553 Graph Databases and Graph Data Visualization

Michal Dupkala

# **Účel programu**

Program umožňuje transformáciu dát zo spravodajského portálu iDnes do grafovej podoby v databázi neo4j.

Transformácií podliehajú:

* nadpis člańku
* autor
* súvisiace články
* témata

Výsledkom sú naloadované dáta, ktoré obsahujú články (node Article), autorov (node Author) a témata (node Topic) so vzťahmi:

* Article -> relates\_to -> Article
* Article -> written\_by -> Author
* Article -> is\_about -> Topic

# Spustenie a používanie

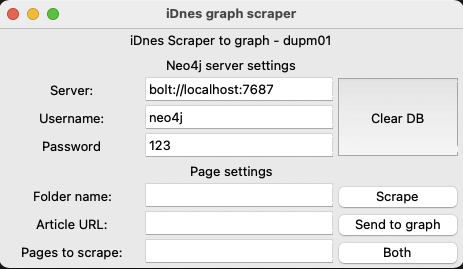
## Požiadavky

1. Nainštalovaný Python 3.9 a vyšší.
2. Nainštalované potrebné balíčky pomocou : *pip install -r /path/to/requirements.txt*
3. Spustená neo4j databáza na ktorú sa môže program napojiť (testované na verzii 4.3.5)

## Spustenie

Program sa spúšťa skriptom bin/main.py . Napríklad pomocou príkazu z root zložky :

*python3 bin/main.py*



Obrázok 1 - Obrazovka programu [zdroj: Autor]

Pre správnu funkcionalitu tlačidla Clear DB, ktoré vymaže všetky uzly a vzťahy z danej databáze je potrebné mať vyplnené všetky položky v časti „Neo4j server settings“.

Tlačidla „Scrape“ , „Send to graph“ a „Both“ fungujú správne pri vyplnení všetkých polí.

## Fungovanie programu a popis kódu

Program je rozčlenený na zložku /conf, ktorá obsahuje konfiguračné súbory o serveri a scrapovaní a na zložku /bin.

V zložke bin sa nachádza main.py, ktorý tvorí grafické rozhranie a funkcionalitu si volá z modulov v zložke /bin/lib – *yaml\_editor.py*, *scraper.py* a*graphtransfer.py* .

### main.py

Hlavný súbor pre spustenie programu.

Definuje grafické rozhranie programu pomocou knižnice tkinter. Rozhranie je rozdelené na 2 stavebné prvky – header a center. V sekcii header nájdeme Label pre nadpis a center je následne rozdelený na top\_row a bottom\_row.

V priestore top\_row sú prvky súvisiace s definíciou a prácou čisto neo4j serverom a v priestore bottom\_row sú prvky pre hlavnú funkcionalitu programu.

Tlačítka sa pomocou lambda funkcií odkazujú a skripty v zložke lib.

|  |
| --- |
| tk.Button(top\_row, text = "Clear DB", command = lambda: [yaml\_editor.edit\_conf\_graph(server\_entry.get(), username\_entry.get(), password\_entry.get()), graphtransfer.clear\_db()], width= 10).grid(row = 1, column = 2, rowspan = 3, sticky = tk.N + tk.S) |

### yaml\_editor.py

Tento modul obsahuje funkcionalitu pre úpravu súborov v /conf. Spúšťa sa po kliknutí tlačidla pred ostatnými súbormi a uloží vstup z textových polí do súborov.

Scraper a graphtransfer k týmto súborom následne pristupujú a berú ich ako vstup.

* conf\_scrape.yaml

url: url na článok

pages\_to\_scrape: počet strániek na zoscrapovanie (do grafu sa dostanú spolu

dir\_name: názov zložky do ktorej sa články uložia v podobe json súborov

* conf\_graph.yaml

server: neo4j server, napíklad: bolt://localhost:7687

username: neo4j

password: password

dir\_name: názov zložky s json súbormi pre vloženie do grafovej databáze

### scraper.py

Súbor scraper.py zabezpečuje scarpovanie. Pre túto funkcionalitu bola zvolená knihovna BeautifulSoup. Zo zdrojového html sú vybrané potrebné informácie a prevedené do json súboru, ktorý sa hneď uloží.

Systém, akými sa stránky postupne scrapujú a ukladajú je nasledujúci:

Uživateľ si zvolí hodnoty „Folder name“ „Article URL“ a „Pages to scrape“.

1. Skript vytvorí v root zložke zložku „Folder name“, pokiaľ už takáto existuje tak sa prepíše.
2. Zoscrapuje prvú stránku a výsledky uloží to „page0.json“.
3. Prečítaj stránky v „references\_to“ v page0.json a zoscrapuj ich. – page1.json, page2.json, page3.json, ....
4. Krok 3 opakuj pre page1.json až pageX.json, kde X = „Pages to scrape“

V priebehu scrapovanie môžeme sledovať proces v termináli:

### graphtransfer.py

Skript pre komunikáciu s neo4j databázov. Použitý bol oficiálne podporovaný [Neo4j Python Driver](https://neo4j.com/developer/python/).

Skript obsahuje triedu Neo4jDB a metódy, ktoré nad ňou vyvárajú operácie pre tvorbu uzlov a vzťahov:

* create\_page\_node – vytvor uzol Article
* create\_author\_node – vytvor uzol Author
* create\_topic\_node – vytvor uzol Topic
* create\_article\_author\_relationship – vytvor vzťah Article WRITTEN\_BY Author
* create\_article\_topic\_relationship - vytvor vzťah Article IS\_ABOUT Author
* create\_article\_article2\_relatinoship - vytvor vzťah Article1 REFERENCES\_TO Article2
* delete\_entire\_db – Vymaž všetky uzly a vzťahy

Všetky tieto funkcie s výjimkou delete\_entire\_db sú volané vo funkcii main, ktorá prechádza json súbory podľa definovaného folder name a vkladá ich do neo4j.

# Ukážka fungovania

V rámci ukážky si vyplníme polia nasledujúcimi údajmi:

Graphical user interface, application

Description automatically generated

Obrázok 2 - Vyplnený program [zdroj: Autor]

Folder\_Name sme si určili Ukazka\_1, article URL ako hlavný článok z 3.12.2021 - <https://www.idnes.cz/zpravy/domaci/valek-covid-koronavirus-nemocnice-nova-vlada.A211203_090038_domaci_ihav> s titulom „Povinné očkování lidí nad 60 let nebude, Anticovid tým nové vlády ho odmítl“ a pages to scrape si nastavíme na 10.

1. Počiatočný stav máme databázu už naplnenú údajmi, preto ju najprv vyčistíme. Tlačítkom Clear DB.

V termináli vidíme, že process prebehol úspešne:



Obrázok 3 - Výpis z konzole, tlačítko Clear DB [zdroj: Autor]

1. V druhom kroku chceme dostať nové údaje do neo4j. Záznamy zatiaľ nemáme uložené v json podobe, preto klikneme tlačidlo both a proces môžeme sledovať v termináli.

Text

Description automatically generated

Obrázok 4 - Výpis z konzole, scraper.py [zdroj: Autor]

V termináli vidíme postupne scrapované články, ktoré odpovedajú sekcií „Související”

Text

Description automatically generated with low confidence

Obrázok 5 - Original souvisejici články [zdroj: Autor]

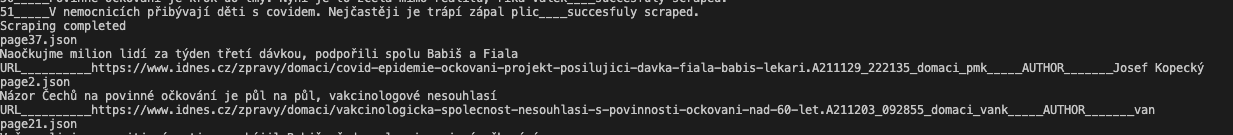
Tieto údajé sa uložili do nové zložky s názvom „Ukázka\_1“.

A screenshot of a computer

Description automatically generated with medium confidence

Obrázok 6 - Novo vytvorená zložka [zdroj: Autor]

1. Po dokončení scrapovania všetkých článkov vidíme v termináli „Scraping completed“ a články sa začnú vkládať do neo4j.



Obrázok 7 - Graphtranfher.py

1. Výsledná naplnená neo4j databáza:

A picture containing chart

Description automatically generated

Obrázok 8 - Naplnená neo4j databáza