1. Spring – Sociomapovanie

Jedná sa o aplikácie na mapovanie interakcií v ľubovoľnom tíme, resp. na mapovanie interakcií medzi tímami. Tá je webová aplikácia, ale má v sebe výpočtový modul.

Problém: výpočtový modul, zodpovedný za správne mapovanie objektov v Euklidovskom priestore. Kritérium kedy výpočet považujeme za dobrý je dosiahnutie istej korelácie medzi maticou vstupných dát a maticou vzdialeností objektov v 2d priestore. Môže sa stať, že požadovanú koreláciu nedosiahneme a mapa tým pádom bude obsahovať nezmysli.

Prečo?

* Vstupné dáta sú príliš náhodné, nedá sa alebo je ťažké ich mapovať
* Mapa sa inicializuje veľmi nešťastne vzhľadom na vstupné dáta.

Všimli sme si, že pri opätovnej spustení výpočtového procesu môžeme dosiahnuť lepšie výsledky.

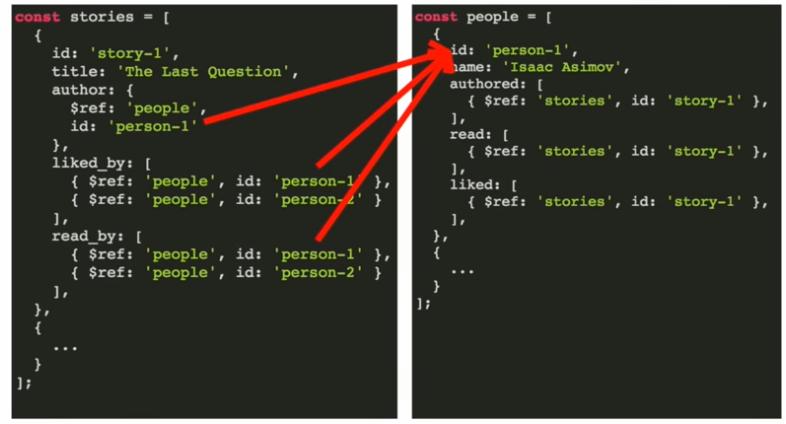
Riešenie:

Na predmete som mal možnosť zoznámiť sa s AOP. Čítal som, že sa dá využiť na ‘retries’. Veľakrát opakovanie danej operácie môže vyriešiť problém ako napr. zavolanie vzdialeného servera (latencia), alebo dopyt na databázu. Cez AOP by som riešil opakované volanie metód – výpočtových, respektíve na zachytenie výnimiek ktoré dostaneme ak sa výpočet neprebehne ako má.

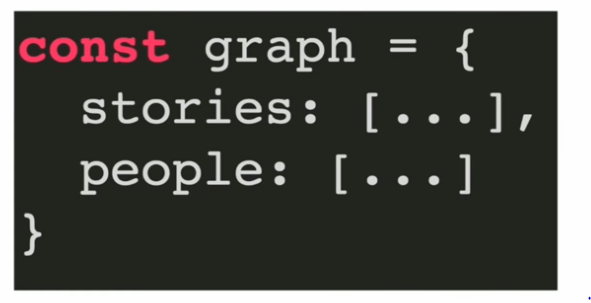
1. Javascript proxy, JSON a Firebase databáza

Na obidvoch Hackathonoch čo som sa zúčastnil používal Firebase databázu. Robil som aj s dátami rovnakými ktoré hneď aj opíšem. Pomocou metódy proxy sa teoreticky dá zjednodušiť prácu, ako som dočítal existujú aj softvérové rámce na tento problém. Treba ho ale používať s rozumom, pozor nato, aby to "nedefinované proxy" nemalo za následok veľmi bolestivé ladenie, pretože skrýva pôvod chýbajúcich údajov a chýb.

Nech máme dáta, ktoré obsahujú referencie na iné objekty (iný JSON).

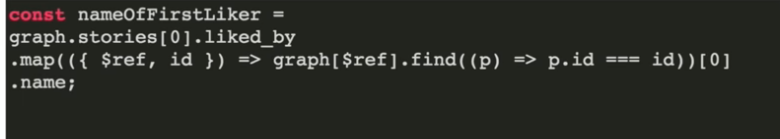


V skutočnosti máme graf. Ak ich spojíme do jedného objektu dostaneme kolekciu stories a people, a tie sú prepojené rôznymi spôsobmi.

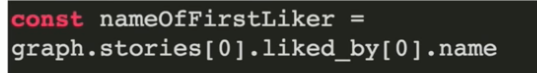


Traverse: povedzme že chceme nájsť človeka, ktorý bol prvý čo lajkol prvý story.

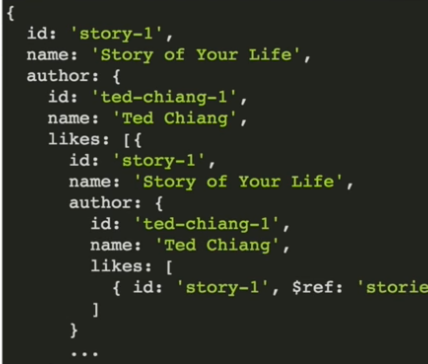
Potrebujeme vytiahnuť všetky referencie, mapovať ich ku objektu a potom až vytiahnúť meno.



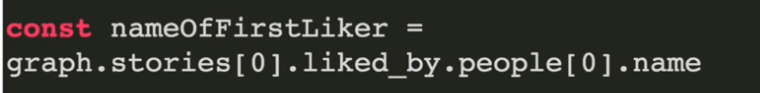
Nebolo by jednoduchšie rovno vytiahnuť meno, keby sa dalo?



Aby to bolo možné chceme mať objekty namiesto referencie. Chceme teda nájsť nejaký spôsob ako transformovať objekty obsahujúce referencie na už vytvorené objekty. Nemôžeme ich vyplniť rekurzívne do nekonečna, lebo nevieme koľko dát budeme mať (eager fetching).



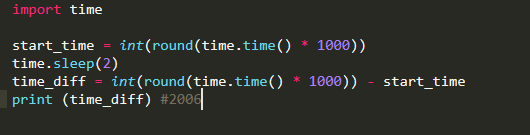
Ale my nemusíme naraz doplniť všetky objekty ale môžeme použiť proxy (lazy fetching). Ako proxy web servery, namiesto referencie mi môžeme presmerovať na daný objekt v druhom, v People kolekcie.



Pomocou proxy sa môžeme tváriť, že máme nekonečne vnorené kolekcie, kým v skutočnosti máme iba 2 na seba sa navážajúce JSON listy.

1. Python, použitie dekorátorov na profiláciu kódu

V Pythone, v čom som viackrát riešil výpočtové procesy profilácia a optimalizácia kódu je častokrát nevyhnutné. Dôležitosť profilácie (časová a pamäťová) zdôrazňuje aj to že veľakrát robím s AWS Lambdou. Je to serverless even-driven computing platforma, platí sa za cash a alokovanú pamäť.



Jasné, dá sa aj tak. Ale to výrazne zvýši počet riadkov kódu a počet volaní.

Odteraz to budem riešiť inak!

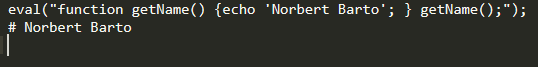


1. PHP –použitie metaprogamovania pri tvorbe WCMS resp. ho využiť pri tvorbe interaktívneho PHP tutoriálu.

Na predmete som si vyskúšal, ako počas behu programu vytvoriť súbor a následne ho použiť.

V rámci svojej bakalárskej práce som vytvoril jeden systém na správu obsahu webu. Mohol som zvýšiť jeho schopnosti pomocou metaprogramovana, resp. zobrať ten WCMS a vytvoriť z neho interaktívny PHP tutoriál – škoda, že momentálne nemám v pláne vrátiť sa ku PHP :/ .

PHP ponúka funkciu, ktorá dokáže vyhodnotiť kúsok kódu PHP reprezentovaného vo formáte reťazca, čo znamená, že je možné, aby jeden užívateľ prijal vstup používateľa alebo prečítal kód zo súboru a získal ho z vášho programu. . Aj iné hotové riešenia používajú tuto technológiu ako napríklad WordPress a online editory kódu



Môžeme použiť file\_get\_contents () na čítanie zdrojového kódu zo súboru a jeho odovzdanie funkcii eval () na spracovanie.

Táto funkcia eval () je veľmi nebezpečná, pretože umožňuje spustenie akejkoľvek formy kódu PHP, takže buďte veľmi opatrní a správne overte všetky vstupy používateľa, ktoré sú odovzdané tejto funkcii.