5 dana na javi 2020 – Challenge zadatak

Košarka predstavlja jedan od najpopularnijih sportova na svetu i u skladu sa potrebama praćenja utakmica i statističkih podataka veliki broj softverskih rešenja se razvija. Takođe spada u nacionalne sportove i samim tim države imaju svoje košarkaške lige.

Trenutno se u svetu teži da svaka liga ima svoj sistem za praćenje rezultata. Kao najjača liga na svetu se smatra Američka NBA liga. Na osnovu njihovog sistema za praćenje utakmica i statistika timova/igrača mi ćemo izgraditi naš prototip.

Plan je da naša aplikacija podrži statističke obrade, kao i rukovanje događajima.

Kao jedno od referentnih rešenja možete pogledati web stranicu https://www.nba.com (na primer: https://www.nba.com/games?date=2020-08-24)

Osnovna košarkaška pravila:

- Na košarkaškoj utakmici učestvuju uvek dva tima.
- Svaki tim je sačinjen od maksimalno 12 registrovanih igrača.
- Utakmica se igra 5 na 5, sto znači da timovi biraju kojih 5 igrača će im se nalaziti na terenu, dok preostalih 7 predstavljaju rezervne igrače.
- Za potrebe zadatka, posmatraćemo da **timovi imaju samo po 5 igrača**, što predstavlja minimum koji je neophodan da bi improvizovali košarkašku utakmicu.
- Pojmovi koje ćemo koristiti u zadatku su: **Koš, Asistencija, Skok**.
- Takođe ovi pojmovi predstavljaju **statističke kategorije** koje će nam biti od koristi.

Slede dalja objašnjenja pojmova:

- **Koš** predstavlja akciju u kojoj igrač ubacuje loptu u koš. Koš može da vredi 1, 2 ili 3 poena. Koševe delimo na koševe postignute iz igre i koševe postignute van igre.
    - 1 poen vredi koš koji je igrač postigao sa linije slobodnog bacanja (nakon prekršaja/faula od strane protivničkog igrača, igrač nad kojim je načinjen faul dobija pravo da šutira jedno ili dva slobodna bacanja kao kompenzaciju za načinjen faul nad njim). Ovaj koš se ne vodi kao koš iz igre, jer prilikom izvođenja slobodnog bacanja protivnički igrači ne smeju da ga čuvaju i igra je zaustavljena.
    - 2 poena nosi koš iz igre sa bilo koje pozicije koja se nalazi unutar zone koja vredi 2 poena.
    - 3 poena nosi koš koji je postignut izvan linije koja označava 3 poena.

- **Asistencija** predstavlja dodavanje (pass) koje je prethodilo košu.
    - Asistencija se pripisuje dodavaču, tj. igraču koji je prosledio loptu do saigrača, koji je nakon tog dodavanja direktno poentirao, odnosno dao koš.
    - Nakon asistencije je moguće postići samo koš iz igre, što znači koš u vrednosti od 2 ili 3 poena.
    - Isti igrač ne može sam sebi asistirati za koš.

- Ako gledamo redosled događaja, nakon asistencije uvek sledi događaj “postignut koš” u vrednosti od 2 ili 3 poena.
- Vrednost same asistencije je uvek 1, ne može biti kao kod koša, koji može da ima vrednost od 1, 2 ili 3 poena.

- **Skok** predstavlja akciju hvatanja lopte nakon promašenog šuta na koš.
  - Kada se šutne na koš i pritom promaši, onaj igrač koji je pokupio loptu nakon tog promašaja dobija skok u svojim statističkim kolonama.
  - Vrednost skoka je uvek 1, ne može biti kao kod koša, koji može da ima vrednost od 1, 2 ili 3 poena.

Zadatak je napraviti Java web aplikaciju za prikaz utakmica i detalja pojedinačnih utakmica.

Potrebno je implementirati, podržati, sledeće upite:

- QUERY_ID:1 - Prikazati sve utakmice sa trenutnim rezultatima. U obzir dolaze i aktivne i završene utakmice.
- QUERY_ID:2 - Prikazati detalje o igračima za izabranu utakmicu, odnosno statistike igrača na toj utakmici (poeni, asistencije, skokovi)
- QUERY_ID:3 - Prikazati statistiku za odabranog igrača za sve utakmice (ukupni i prosek poena, skokova, asistencija)
- QUERY_ID:4 – Prikazati igrače koji su postigli najviše poena, skokova i asistencija na nivou svih utakmica (jedan igrač po kategoriji, ukoliko je više igrača postiglo isti maksimalni broj prikazati ih sve)
- QUERY_ID:5 – Prikazati top 5 igrača sa najviše “double-double”-ova (bilo koje 2 statističke kategorije za koje je igrač postigao dvocifren učinak. Pojašnjenje u nastavku zadatka)
- QUERY_ID:6 - Prikazati rangiranje timova – izračunava se na osnovu procenta pobeda u svim utakmicama (ukoliko timovi imaju isti procenat, prednost dobija onaj tim koji ima bolju ukupnu koš razliku)

Dodatna pojašnjenja za upite:

- **Double-double** na utakmici ostvaruje igrač koji je postigao dvocifren broj poena u dve različite statističke kategorije. U praksi to može biti miks bilo koje dve kategorije.

  Primer 1: 10+ koš poena i 10+ asistencija
  Primer 2: 10+ koš poena i 10+ skokova
  Primer 3: 10+ asistencija i 10+ skokova
  Ukoliko igrač u sve tri kategorije ime dvocifren broj poena, taj pojam se naziva triple-double. Ovaj podatak nećemo posebno obrađivati, te ćemo ovakvu situaciju takođe posmatrati kao double-double pošto je on podskup od triple-double učinka.

- **Ukupna "koš razlika"** po timu predstavlja razliku između ukupnog broja postignutih koš poena na svim utakmicama i ukupnog broja primljenih koš poena. (suma datih poena - suma primljenih poena)

Svi upiti treba da budu implementirani kao REST endpoint-ovi. Rezultate upita QUERY_ID:1 i QUERY_ID:2 treba prikazati na frontend-u.

Frontend:

Inicijalni frontend treba da sadrži sledeće vizualizacije (QUERY_ID:1, QUERY_ID:2):

- Prikaz svih utakmica sa rezultatima

Los Angeles Lakers - Miami Heat	
Los Angeles Lakers	**106**
Miami Heat	93
more details	

Orlando Magic - Houston Rockets	
Orlando Magic	**99**
Houston Rockets	94
more details	

Golden State Warriors - Chicago Bulls	
Golden State Warriors	85
Chicago Bulls	**89**
more details	

Los Angeles Lakers - Chicago Bulls	
Los Angeles Lakers	**101**
Chicago Bulls	100
more details	

- Prikaz detalja jedne odabrane utakmice sa statistikama igrača na toj utakmici (broj koševa, asistencija i skokova po igraču)

Los Angeles Lakers - Miami Heat

Player name	Points	Assists	Jumps
LeBron James	20	10	12
Dwight Howard	14	13	3
Jimmy Butler	31	6	7

Treba imati na umu da se u narednoj fazi očekuje proširenje frontend-a.

Ulaz u aplikaciju su:

- Konfiguracioni JSON fajl sa podacima o timovima i igračima
- JSON fajl sa listom event-ova koji opisuju tok utakmica

Prilikom učitavanja JSON fajl-a sa podacima, preskočiti učitavanje event-ova koji ne zadovoljavaju pravila (pravila/constaint-ovi su opisani kasnije u dokumentu) i logovati ih.

Izlaz iz aplikacije treba da sadrži podatke definisane odabranim upitom. Format ispisa je JSON.

Struktura ulaznih fajlova:

- Konfiguracioni fajl je u JSON formatu, gde jedan JSON objekat prestavlja 1 unos.
- Fajl sa event-ovima je u JSON formatu, gde je sadržaj fajla JSON array i svaki unos u nizu predstavlja jedan event.

Primeri ulaznih fajlova se nalaze na lokaciji: http://www.5dananajavi.com/wp-content/uploads/2020/11/5-dana-na-javi-Challenge-fajlovi-2020.zip

U datom zip fajlu se nalaze sledeća 4 JSON fajla:

- teams.json – konfiguracioni fajl sa svim timovima
- players.json – konfiguracioni fajl sa svim igračima
- events.json – JSON fajl sa listom event-ova koji opisuju tok utakmica
- events_with_errors.json - JSON fajl sa listom event-ova koji opisuju tok utakmica, ali koji sadrži i nevalidne event-e koje ne treba učitavati

Nije potrebno učitavati obe liste event-ova istovremeno, već učitavati “events_with_errors” nakon što implementirate validaciju.

Prilikom ocenjivanja koristiće se drugi set podataka, ali će format i struktura fajlova biti ista.

Lista definisanih JSON objekata:

Team:

```
{

      Id: long,
      name: string,
      city: string

}
```

Player:

```
{
      Id: long,
      teamId: long,
      name: string,
      surname: string,
      number: int,
      height: int,
      age: int,
      position: enum
}
```

Polje “position” je tipa ENUM i prihvata sledeće vrednosti:

1. POINT_GUARD
2. SHOOTING_GUARD
3. SMALL_FORWARD
4. POWER_FORWARD
5. CENTER

Event:

```
{
      game: long,
      type: enum,
      payload: object
}
```

Payload:

```
{
      hostId: long,
      guestId: long,
      playerId: long,
      value: int
}
```

Polje “type” je tipa ENUM i prihvata sledeće vrednosti:

1. START
2. END
3. ASSIST
4. JUMP
5. POINT

Pojašnjenje event-ova:

- START – označava početak utakmice, predstavlja prvi event za utakmicu
- END – završetak utakmice, nakon ovog događaja ne mogu se dodavati novi event-ovi za utakmicu
- POINT - koš
- ASSIST - asistencija
- JUMP - skok

Polja u “Payload” JSON objektu zavise od vrste tipa Event-a u kom se nalaze.

START event: treba da sadrži hostId i guestId, odnosno id-jeve domaćeg i gostujućeg tima

END event: ne treba da sadrži ni jedan atribut iz payload objekta

POINT/ASSIST/JUMP event-ovi: treba da sadrže playerId i value, odnosno id igrača i vrednost event-a koji je postigao.

Primeri:

- Za koš, vrednost atributa value može da bude 1, 2 ili 3 (min 1 / max 3)
- Za asistenciju, vrednost atributa value može da bude samo 1 (min 1 / max 1)
- Za blok, vrednost vrednost atributa value može da bude takođe samo 1 (min 1 / max 1)

Event constraints:

- Završena utakmica mora da bude obuhvaćena između eventova START i END.
- Takođe je bitno naglasiti mogućnost nedostatka END event-a, što samim tim znači da je utakmica još uvek aktivna, odnosno da je u toku.
- Nakon događaja ASSIST može da se desi jedino događaj POINT (koš) i to u vrednosti od 2 ili 3 poena.
- Za event-ove POINT i JUMP nemamo veća ograničenja i oni mogu da se desašavaju u bilo kom redosledu i da se ponavljaju sami za sobom.
- Igrač kom se dodaje bilo koji tip event-a mora da bude član tima koji učestvuje na toj utakmici.

Tehnologije:

- Obavezno: Java 8
- Obavezno: korišćenje build alata (Maven ili Gradle)
- Obavezno: rešenje zadatka mora da radi sa opensource free software rešenjima
- Preporučeno: Spring framework / Spring Boot (ali se može koristiti bilo koji open source framework, ukoliko ga želite koristiti)

Rešenje treba da sadrži:

- izvorni kod
- fajlove korišćene za testiranje
- dokumentaciju:
    - opis okruženja potrebnog da se uradi build
    - kako se radi build
    - primer kako se aplikacija pokreće
    - primer pozivanja REST endpoint-ova za svaki implementiran query
    - listu korišćenih tehnologija sa kratkim opisom
    - dijagram klasa koje opisuju model podataka

Svaki prijavljeni takmičarski tim dobija korisnički nalog na GitLab-u, gde je potrebno kreirati repozitorijum koji će sadržati rešenje zadatka.

Rešenjem se smatra sadržaj kreiranog Git repozitorijuma 26. novembra u 9 časova.

Ukoliko postoje nejasnoće u vezi zadatka, slobodno pošaljite pitanja na e-mail na adresu pitanja@5dananajavi.com ili koristeći formu za kontakt na sajtu takmičenja https://www.5dananajavi.com

Sva primljena pitanja i odgovori moći će se pronaći na sajtu takmičenja i biće osvežavani tokom celog kvalifikacionog perioda.