Zadatak 5 dana u oblacima 2024

Opis funkcionalnog zahteva

Gejming je postao mnogo više od običnog hobija – danas je to globalna industrija u svetu zabave, jedna od najbrže rastućih. Uz rast zajednice i kulture koja se stvara oko igara, gejming predstavlja jedinstveni spoj socijalne interaktivnosti, umetnosti i tehnologije. Sa rastom gejminga povećava se i potreba za gejming platformama i zajednicama radi razmena iskustava i podsticanja veština.

Popularna gejming kompanija je kontaktirala Levi9 radi izrade matchmaking platforme za igre. Prva faza ovog softvera uključuje izradu API-ja za matchmaking za njihovu novu FPS igru koja treba da na osnovu odigranog meča izračuna rejting (ELO) i dodatne statistike za svakog igrača koji je učestvovao u tom meču na osnovu zadatih podataka.

Opis zadatka

Potrebno je implementirati aplikaciju (Java, Python, C# ili Node.js) koja omogućava upis novih igrača, timova i mečeva u bazu podataka. Takođe mora da računa ELO i ostale statistike za svakog igrača prilikom kreiranja meča i vraća detalje o igračima i njihovim statistikama.

API

Players:

Kreiranje igrača:

POST /players/create

Request:

```
{
  "nickname": "Player1"
}
```

Response:

```
{
  "id": "b769b730-d1b9-4a94-8d2d-2936e050722c",
  "nickname": "Player1",
  "wins": 0,
  "losses": 0,
  "elo": 0,
  "hoursPlayed": 0,
  "team": null,
  "ratingAdjustment": null
}
```

Validacija:

- nickname mora biti jedinstven
- u slučaju da validacija ne prolazi vratiti response code različit od 2xx

Pronalaženje playera po ID-ju:

GET/players/b769b730-d1b9-4a94-8d2d-2936e050722c

Response:


```
{
  "id": "b769b730-d1b9-4a94-8d2d-2936e050722c",
  "nickname": "Player1",
  "wins": 0,
  "losses": 0,
  "elo": 0,
  "hoursPlayed": 0,
  "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
  "ratingAdjustment": 50
}
```


- Ukoliko igrač nije pronađen vratiti 404

Teams:

Kreiranje tima:

POST /teams

Request:


```
{
 "teamName": "Team1",
 "players": [
   "b769b730-d1b9-4a94-8d2d-2936e050722c",
   "34e7a9e7-df16-4082-92d5-cadb8fc087e5",
   "64f6186b-3a39-47b9-9313-3fcb8e4f06fd",
   "187b683c-1255-46a6-a709-60f8af0fd200",
   "f8987880-8869-472c-8335-83f60132b15d"
 ]
}
```


Validacije:

- svaki tim mora imati tačno 5 igrača
- teamName mora biti jedinstven
- jedan igrač može da bude samo u jednom timu
- igrac ne može da menja timove (nakon inicijalnog dodavanja u tim)
- u slučaju da neki od igrača nije pronađen vratiti 4xx
- u slučaju da validacija ne prolazi vratiti response code različit od 2xx

Response:


```
{
 "id": "b909d79d-04d3-442d-9b43-29b2a44cc628",
```

```
"teamName": "Team1",
"players": [
    {
        "id": "f8987880-8869-472c-8335-83f60132b15d",
        "nickname": "Player5",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50

    },
    {
        "id": "34e7a9e7-df16-4082-92d5-cadb8fc087e5",
        "nickname": "Player2",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
    },
    {
        "id": "187b683c-1255-46a6-a709-60f8af0fd200",
        "nickname": "Player4",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
    },
    {
        "id": "b769b730-d1b9-4a94-8d2d-2936e050722c",
        "nickname": "Player1",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
    },
    {
```

```
        "id": "64f6186b-3a39-47b9-9313-3fcb8e4f06fd",
        "nickname": "Player3",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
    }
  ]
}
```


Pronalaženje tima po ID-ju:

GET /teams/b909d79d-04d3-442d-9b43-29b2a44cc628

Response:


```
{
  "id": "b909d79d-04d3-442d-9b43-29b2a44cc628",
  "teamName": "Team1",
  "players": [
    {
      "id": "b769b730-d1b9-4a94-8d2d-2936e050722c",
      "nickname": "Player1",
      "wins": 0,
      "losses": 0,
      "elo": 0,
      "hoursPlayed": 0,
      "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
      "ratingAdjustment": 50
    },
    {
      "id": "34e7a9e7-df16-4082-92d5-cadb8fc087e5",
      "nickname": "Player2",
      "wins": 0,
      "losses": 0,
      "elo": 0,
      "hoursPlayed": 0,
      "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
      "ratingAdjustment": 50
    },
    {
      "id": "64f6186b-3a39-47b9-9313-3fcb8e4f06fd",
      "nickname": "Player3",
      "wins": 0,
      "losses": 0,
```

```
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
      },
      {
        "id": "187b683c-1255-46a6-a709-60f8af0fd200",
        "nickname": "Player4",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
      },
      {
        "id": "f8987880-8869-472c-8335-83f60132b15d",
        "nickname": "Player5",
        "wins": 0,
        "losses": 0,
        "elo": 0,
        "hoursPlayed": 0,
        "teamId": "b909d79d-04d3-442d-9b43-29b2a44cc628",
        "ratingAdjustment": 50
      }
    ]
}
```


- ukoliko tim ne postoji vratiti 404

Matches:
Dodavanje meča:
POST /matches
Request:

```
{
   "team1Id": "dacfe004-42d8-4938-8e1c-a1fe46739cb6",
   "team2Id": "7265bc21-46bc-40e3-a2d5-3338c8cc7495",
   "winningTeamId": "dacfe004-42d8-4938-8e1c-a1fe46739cb6",
   "duration": 3
}
```

- team1Id, team2Id i winningTeamId id-jevi odgovarajućih timova
- ukoliko se meč završio nerešenim rezultatom proslediti null za winningTeamId, wins i losses u ovom slučaju ne menjati
- duration je ceo broj i to dužina trajanja meča u satima i dodaje se na ukupno vreme igre (hoursPlayed) za svakog igrača. Duration ne sme biti manji od 1 prilikom unosa
- u ovom koraku treba izračunati ELO igrača, hoursPlayed, broj pobeda i broj poraza
- u slučaju da validacija ne prolazi vratiti response code različit od 2xx

Response 200 OK:

Raspored unosa podataka:
Prvo je potrebno dodati igrače, dodati ih u timove i tek nakon toga treba dodavati mečeve.

Statistika

ELO:

Ukoliko igrač pobedi S = 1
Ukoliko igrač izgubi S = 0
Nerešeno S = 0.5
R1 je trenutna ELO vrednost igrača za kog računamo novi ELO
R2 je prosečna ELO vrednost suprotnog tima

Formula računanja očekivanog ELA:

$$E = \frac{1}{1 + 10^{(R2-R1)/400}}$$

Formula računanja nove vrednosti za ELO igrača:

$$Rnew = R1 + K \times (S - E)$$

Konstanta K (ratingAdjustment) je konstanta koja određuje koliko brzo se rejting menja nakon svakog meča i može biti 10, 20, 30, 40 ili 50. Novi igrači obično počinju sa većim vrednostima dok se ne stabilizuje ranking pa se ta konstanta vremenom smanjuje.
Konstanta K se određuje po sledećoj logici:

- Ukoliko igrač ima manje od 500h, k=50
- Ukoliko igrač ima između 500h i 999h, k=40
- Ukoliko igrač ima između 1000h i 2999h, k=30
- Ukoliko igrač ima između 3000h i 4999h, k=20
- Ukoliko igrač ima 5000h ili više, k=10

Broj odigranih sati igrača (h) se računa tako što se na trenutnu vrednost (hoursPlayed) doda dužina trajanja meča (duration).

Tehnički zahtevi

Aplikacija mora biti implementirana u jednoj od navedenih tehnologija: Java, Python, C#, Node.js.

Obavezno koristiti build alate/package manager-e, kao što su Maven/Gradle, Pip, NPM/Yarn, NuGet.

Obavezno je koristi in-memory bazu.

Rešenje mora u potpunosti biti zasnovano na opensource (free) software-u.

Izvorni kod rešenja mora biti upload-ovan na GitHub, kao zaseban repozitorijum. Repozitorijum treba da sadrži:

- Izvorni kod
- Dokumentaciju u Readme.md fajlu:
    - Opis okruženja potrebnog da se uradi build
    - Kako se radi build
    - Primer kako se aplikacija pokreće
    - Listu korišćenih tehnologija sa kratkim opisom
- Na repozitorijum je potrebno dodati sledeće osobe kao kolaboratore (podešavanje se vrši preko Settings / Collaborators and Teams / Add people):
    - ugmizic
    - Galic-Miroslav
    - p-djulinac
    - MilovanovicA
    - dkasalica
    - teodoramandiclevi9
    - draganakuzminac85
    - milicamedic153
    - dkrickovic
    - predragdraganovic
    - predragIvancevic97
    - jovan2k1

Ocenjivanje i bodovanje

Prilikom ocenjivanja rešenja, biće uzeti u obzir i funkcionalni i tehnički aspekt rešenja, odnosno i to da li aplikacija zaista radi što se očekuje, kao i to na koji način je rešenje implementirano, dokumentovano, itd.

Funkcionalni zahtevi – max 60 bodova

- Javni test case – max 20 bodova
- Ostali test case-ovi – max 40 bodova

Implementacija – max 40 bodova

- Kvalitet rešenja (arhitektura projekta, organizacija koda, čitljivost itd) – max 30 bodova
- Unit testovi – max 5 bodova
- Readme – max 5 bodova – objašnjenje na koji način se build-a i pokreće projekat

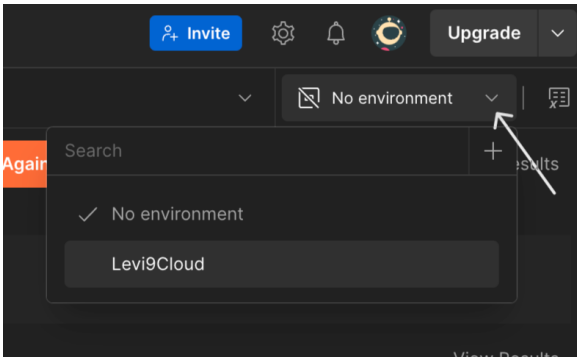
Javni test case

Pored ovog dokumenta svako od vas će dobiti Postman kolekciju i environments fajl koji su potrebni za pokretanje javnog test case-a.
URL za skidanje Postmana: https://www.postman.com

Nakon pokretanja Postmana uvucite kolekciju i environment fajlove. File -> Import i izaberite oba fajla sa .json ekstenzijom.

Nakon importa proverite da li vam se u collections i environments delu nalaze importovana kolekcija i environment varijabla u sekcijama Collections i Environments.

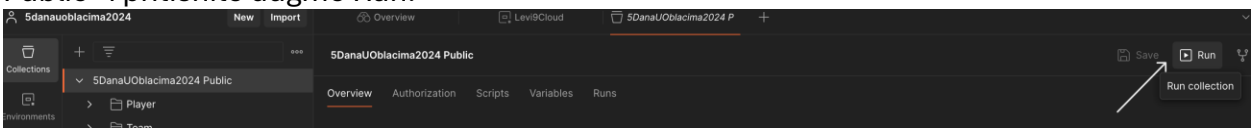
Nakon toga izaberite importovani environment klikom na sledeće dugme i izaberite Levi9Cloud:



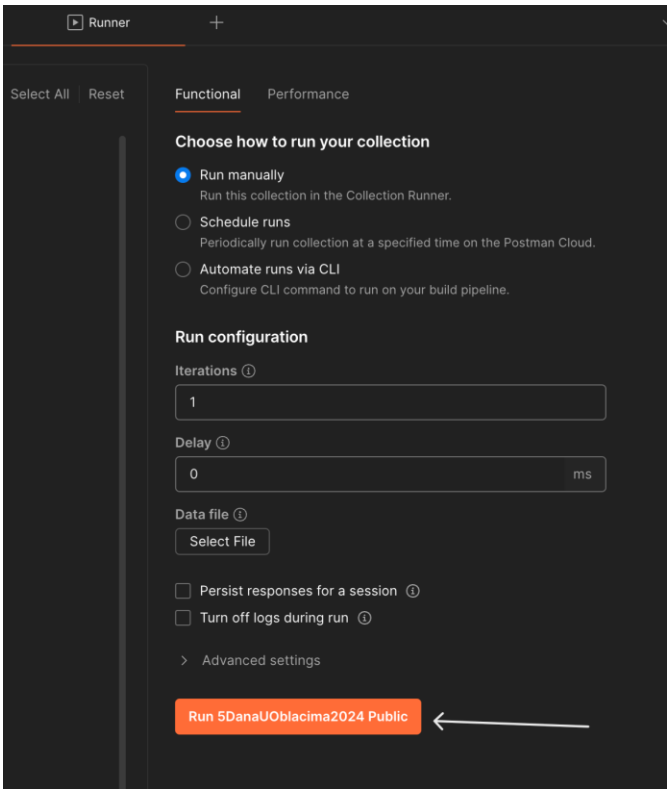
Ukoliko vam aplikacija trči na portu koji nije 8080, promenite baseURL varijablu u environments.

U trenutku pokretanja testova baza mora biti prazna.

Da bi ste pokrenuli testove u Collections kliknite na kolekciju “5danauoblacima2024 Public” i pritisnite dugme Run.



Nakon toga kada se otvori prompt za pokretanje testova kliknite na dugme “Run 5danauoblacima 2024Public”.



Testovi koji su prošli biće obeleženi zelenom bojom a testovi koji su pali crvenom.