

# Dokumentacija

All games

## 5 dana na Javi

Sanja nas organizovala

### Games

Game Id	Host Name	Guest Name	Host Score	Guest Score
1	Charlotte Hornets	Brooklyn Nets	105	122
2	Atlanta Hawks	Boston Celtics	92	105
3	Boston Celtics	Brooklyn Nets	74	106
4	Charlotte Hornets	Atlanta Hawks	102	97
5	Brooklyn Nets	Boston Celtics	90	86
6	Charlotte Hornets	Atlanta Hawks	76	87

### Game stats

First Name	Last Name	Points	Assists	Rebounds
------------	-----------	--------	---------	----------

### Opis okruzenja potrebnog da se uradi build:

Okruzenja koja su nam potrebna kako bismo uradili build su: Java Spring, Java Spring Boot

### Kako se radi build:

Aplikacija se builduje tako sto se pokrene mvn clean install.

### Primer kako se pokrece aplikacija:

Aplikacija se pokrece kroz klasu Application, koja se nalazi u backend folderu. Nakon sto smo pokrenuli aplikaciju, ona pocinje da radi na localhost-u na portu 8080. Na stranici se nalazi rezultat query1, a kada kliknemo na neku utakmicu (red u tabeli), u novoj tabeli se prikazuje statistika za sve igrace na izabranoj utakmici.

Takodje fajl koji loguje nevalidne događaje log.txt se nalazi u korenu projekta. Fajlovi koji se koriste za ulaz u program se nalaze u resources folderu sa imenima teams.json, players.json i events.json.

### Primer pozivanja REST endpoint-ova za svaki implementiran query:

1. query1: get zahtev saljemo na putanji localhost:8080/api/1. Kao povratnu vrednost

dobijamo listu svih utakmica gde svaka utakmica ima sledece podatke: naziv domaceg tima, naziv gostujuceg tima, broj postignutih poena domacina, broj postignutih poena gosta, da li je utakmica zavrшена i id utakmice.

2. query2: get zahtev saljemo na putanji localhost:8080/api/2/id. Kao povratnu vrednost dobijamo listu svih igrača jedne utakmice, a to je utakmica čiji smo id prosledili u putanji. Dobijamo ucinak svakog igrača na toj utakmici, a to je broj postignutih poena, asistencija i skokova.
3. query3: get zahtev saljemo na putanji localhost:8080/api/3/id. Kao povratnu vrednost dobijamo podatke o igraču čiji smo id prosledili u putanji. Dobijamo broj ukupno postignutih poena tog igrača, prosek poena po utakmici, broj ukupnih asistencija, prosek asistencija po utakmici, broj ukupnih skokova, prosek skokova po utakmici.
4. query4: get zahtev saljemo na putanji localhost:8080/api/4. Kao povratnu vrednost dobijamo igrača koji ima najbolji skor u određenoj kategoriji kao i koliki je taj skor.
5. query5: get zahtev saljemo na putanji localhost:8080/api/5. Kao povratnu vrednost dobijamo top 5 igrača koji su postigli najviše "double-double"-ova i koliko su postigli "double-double"-ova.
6. query6: get zahtev saljemo na putanji localhost:8080/api/6. Kao povratnu vrednost dobijamo sortirane timove po procentu ostvarenih pobeda, ako je procenat pobeda isti onda se rangiraju po kos razlici. Imamo podatke o tome koliki je procenat pobeda i kolika je kos razlika.

### Lista koriscenih tehnologija sa kratkim opisom:

Za izradu backend dela projekta je koriscen Java Spring Boot framework, a za izradu frontend dela projekta je koriscen Vue JS framework. U backendu u klasi Controller imamo putanje za get zahteve, a zatim u frontendu uzimamo te podatke koje smo poslali i prikazujemo u browseru. Nismo koristili nikakvu bazu podataka vec podatke uzimamo iz json fajlova. Citanje iz fajlova radimo u klasi StorageReader.

### Dijagram klasa koje opisuju model podataka:

