

Web based systems NBA viewer

Марко Марковиќ

Факултет за информатички науки и компјутерско инженерство – ФИНКИ

Веб Базирани Системи

Милош Јовановиќ

Осврт

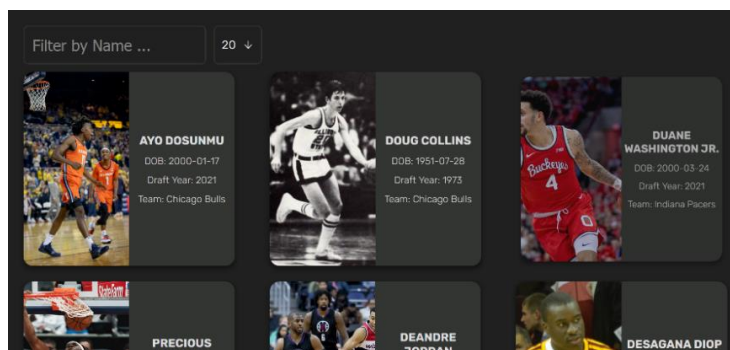
Проектот е веб-апликација напишана користејќи *'React'* како главна алатка за рендерирање и обработка на податоците. Направена е со цел да се манипулира со податоци добиени преку праќање на *SPARQL* барања. Множеството на податоци се наоѓа на dbpedia.org, додека пак самите барања се праќаат до крајната точка [тука](#). Темата на апликацијата е прелистувач за сите NBA играчи кои се наоѓаат како ресурси во множеството. Постојат опции за филтрирање на резултатите како и посебен преглед за секој од играчите присутни на страната. Самата апликација може да ја видите [тука](#). Доколку пак ве интересира структурата и како е напишана апликацијата можете да го посетите овој [линк](#).

Сервиси и Компоненти

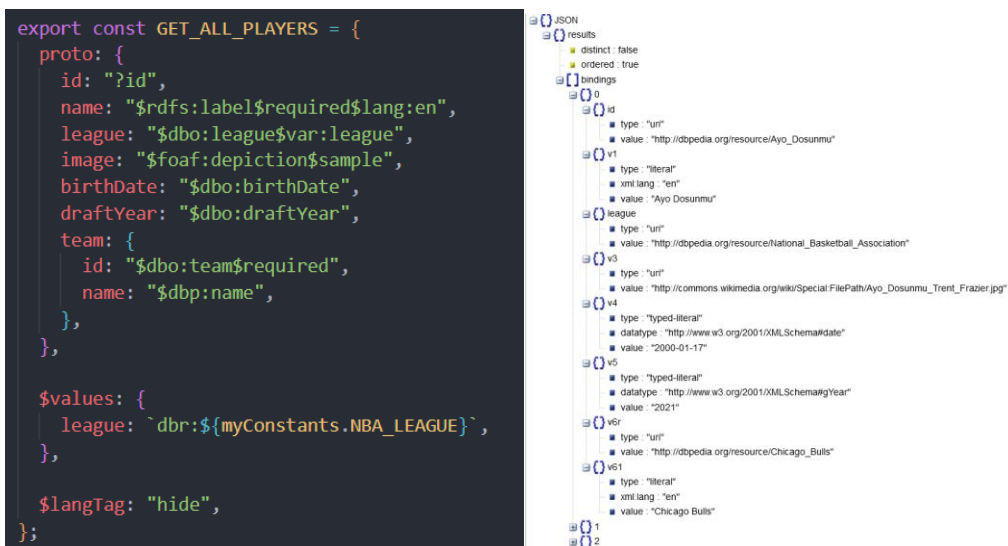
Најзначајниот дел од апликацијата е сервисот кој се користи за добивање на податоците. Тој е дефиниран во *'SparqlFactory.jsx'* датотеката. Тој содржи два повици коишто примаат *'квериња'* како параметри. Тие се напишани во форма на *JSON* објекти и се дефинирани како константи во самата апликација. Трансформацијата на овие *'квериња'* во валидни *SPARQL* барања е овозможено од страна на надворешната библиотека *'SparqlTransformer'*. За самата библиотека и начинот како работи истата можете да го отворите овој [линк](#).

Почетната страна на апликацијата го користи главното *'квери'* - *'GET_ALL_PLAYERS'*. Овој повик резултира во низа од објекти која е всушност сите играчи со сите нивни атрибути како слика, наслов, датум на раѓање и слично. Атрибутите за секој објект потоа се праќаат на посебна компонента *'PlayerCard.jsx'* како својства.

На оваа страна исто така има посебна логика за да се сочува променлива во прелистувачот која ги памти добиените резултати. Ова е со цел корисникот да има подобро искуство и да не мора да чека повторно да се прочитаат податоците при вчитување на страната повеќе пати. На горниот дел од страната се наоѓаат две влезни полиња за филтрирање на резултатите. Овие елементи си имаат своја имплементација дефинирана во *'Header.jsx'* и *'CustomSelect.jsx'*.



Слика 1. Почетна страна



Слика 2. Главно квери и пример резултат

При притискање на некоја од картичките за секој од играчите апликацијата рендерира уникатна *'Details.jsx'* компонента која ги прикажува сите атрибути за тој играч. На оваа страна се праќа втор повик кој го содржи клучот на играчот како посебен параметар. Клучот се праќа на компонентата при кликање на картичката од почетната страна. Тука има помошни анимации кои го подобруваат корисничкото искуство се со цел корисникот да не гледа празна содржина додека се добијат резултати од повикот. Компонентата различно се рендерира во зависност од клучот препратен. Доколку корисникот се обиде да пристапи до непостоечка страна или пак по грешка се препрати погрешен клуч апликацијата рендерира соодветна *'NotFound.jsx'* компонента.



Слика 3. Преглед на играч

```
export const GET_PLAYER = (playerId) => {
  return {
    proto: {
      id: "?id",
      name: "$rdf:type$required$lang:en",
      image: "$foaf:depiction$sample",
      birthDate: "$dbo:birthDate",
      draftYear: "$dbo:draftYear",
      team: {
        id: "$dbo:team$required",
        name: "$dbp:name",
      },
      position: {
        id: "$dbo:position$required",
        name: "$rdf:type$required$lang:en",
      },
      abstract: "$dbo:abstract$required$lang:en",
      number: "$dbo:number",
    },
    $values: { "?id": "${playerId}" },
    $langTag: "hide",
  };
};
```

Слика 4. Квери за посебен играч

Стилови

Изгледот на апликацијата се состои од обичен CSS вклучувајќи неколку варијабли за боите искористени со цел да има дизајнот некаква конзистентност. Тука битно е да се спомене дека се посвети време на тоа дизајнот да е пристапен и ‘responsive’ за корисникот да има оптимално искуство. Сите стилови за различните страни/компоненти се дефинирани во [‘styles’](#) директориумот.

Тестирање

Бидејќи апликацијата е мала, одлучено беше да се покрие функционалноста со неколку ‘e2e’ сценарија кои би го симулирале однесувањето на некаков корисник. Напишани се неколку тестови користејќи ‘Cypress’, модерна алатка за креирање на брзи и ефикасни тестови. Тестовите се интуитивни и дескриптивни. Во суштина го покриваат рендерирањето на резултатите, филтрирањето на истите и еднаквоста помеѓу податоците прикажани на почетната страна за одреден играч со се оние кои се наоѓаат на посебниот преглед за тој играч. Можете да видите како се напишани тестовите [овде](#).

Користена литература

[‘Transforming the JSON Output of SPARQL Queries for Linked Data Clients’](#) - Pasquale Lisena, Raphaël Troncy