



## ReactJS - reaktivna arhitektura

# Parcijalni ispit

U sklopu parcijalnog ispita, potrebno je napraviti jednostavnu React aplikaciju koja se sastoji od dva dijela:

1. Forma (obrazac) za unos korisničkog imena
2. Prikaz detalja unešenog korisnika

Kao bazu korisnika, koristit ćemo GitHub (<https://github.com>). GitHub je servis na kojem se nalazi velik broj programskih repozitorija. Npr:

- React: <https://github.com/facebook/react>
- Redux: <https://github.com/reduxjs/redux>
- React-Bootstrap: <https://github.com/react-bootstrap/react-bootstrap>

Svaki repozitorij ima svog vlasnika. Svaki korisnik na GitHubu može imati mnogo repozitorija. Npr. ovo je pregled svih repozitorija koje posjeduje korisnik Facebook: <https://github.com/facebook>.

### Forma za unos korisničkog imena:

Forma se sastoji od jednog input polja i jednog buttona.

U input polje potrebno je upisati korisničko ime bilo kojeg GitHub korisnika - npr. "facebook".

Korisničko ime GitHub korisnika moguće je pročitati iz bilo kojeg GitHub url-a.

Primjer:

- <https://github.com/facebook/react> - korisničko ime = *facebook*
- <https://github.com/reduxjs/redux> - korisničko ime = *reduxjs*
- <https://github.com/react-bootstrap/react-bootstrap> - korisničko ime = *react-bootstrap*

Po unosu GitHub korisničkog imena, na klik buttona, potrebno je s GitHub REST API-ja dohvatiti i prikazati detalje tog korisnika i njegove repozitorije.



Primjer kako forma može izgledati:

GitHub username:

GO!

## GitHub REST API

GitHub API nalazi se na sljedećem URL-u: <https://developer.github.com/v3/repos/>.  
Za uspješno odrađivanje ove vježbe, bitna su dva poziva:

### 1. Dohvaćanje GitHub usera

Dohvaćanje GitHub usera moguće je napraviti GET pozivom na sljedeći URL:  
<https://api.github.com/users/<korisničko-ime>>.

Npr:

- Za dohvaćanje detalja korisnika s korisničkim imenom *facebook*, potrebno je napraviti GET poziv na URL <https://api.github.com/users/facebook>
- Za dohvaćanje detalja korisnika s korisničkim imenom *reduxjs*, potrebno je napraviti GET poziv na URL <https://api.github.com/users/reduxjs>



Primjer API odgovora za poziv <https://api.github.com/users/reduxjs>:

```
{
  "login": "reduxjs",
  "id": 13142323,
  "node_id": "MDEyOk9yZ2FuaXphdGlvbWZMTQyMzIz",
  "avatar_url": "https://avatars3.githubusercontent.com/u/13142323?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/reduxjs",
  "html_url": "https://github.com/reduxjs",
  "followers_url": "https://api.github.com/users/reduxjs/followers",
  "following_url": "https://api.github.com/users/reduxjs/following{/other_user}",
  "gists_url": "https://api.github.com/users/reduxjs/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/reduxjs/starred{/owner}/{repo}",
  "subscriptions_url": "https://api.github.com/users/reduxjs/subscriptions",
  "organizations_url": "https://api.github.com/users/reduxjs/orgs",
  "repos_url": "https://api.github.com/users/reduxjs/repos",
  "events_url": "https://api.github.com/users/reduxjs/events{/privacy}",
  "received_events_url": "https://api.github.com/users/reduxjs/received_events",
  "type": "Organization",
  "site_admin": false,
  "name": "Redux",
  "company": null,
  "blog": "https://redux.js.org/",
  "location": "The Moon",
  "email": null,
  "hireable": null,
  "bio": "Redux is a predictable state container for JavaScript apps.",
  "public_repos": 12,
  "public_gists": 0,
  "followers": 0,
  "following": 0,
  "created_at": "2015-07-01T23:08:35Z",
  "updated_at": "2018-04-24T14:52:24Z"
}
```

Iz ovog odgovora potrebno je izdvojiti i prikazati sljedeće informacije o korisniku:

- **avatar\_url** - url link na profilnu sliku korisnika
- **name** - ime korisnika
- **location** - lokacija korisnika
- **bio** - detalji korisnika

## 2. Dohvaćanje repozitorija GitHub usera

Dohvaćanje repozitorija GitHub usera moguće je napraviti GET pozivom na sljedeći URL: <https://api.github.com/users/<korisničko-ime>/repos>.

Npr:

- Za dohvaćanje repozitorija korisnika s korisničkim imenom *facebook*, potrebno je napraviti GET poziv na URL <https://api.github.com/users/facebook/repos>
- Za dohvaćanje repozitorija korisnika s korisničkim imenom *reduxjs*, potrebno je napraviti GET poziv na URL <https://api.github.com/users/reduxjs/repos>




Odgovor na ovaj API poziv je JSON lista repozitorija navedenog korisnika (kako biste vidjeli puni odgovor s GitHub API-ja, otvorite link u internet pregledniku). Svaki element liste ima sljedeće bitne atribute:

- **id** - jedinstveni broj koji predstavlja identifikator GitHub repozitorija
- **name** - ime repozitorija

### Prikaz detalja unešenog korisnika:

Po unosu korisničkog imena, klikom na gumb, dohvaćaju se s REST API-ja i prikazuju detalji GitHub korisnika.

Primjer kako možemo prikazati korisničke detalje:



## Redux

**BIO:** Redux is a predictable state container for JavaScript apps.

**LOCATION:** The Moon

**REPOSITORIES:**

d3-state-visualizer
react-redux
react-redux-benchmarks
redux
redux-devtools
redux-devtools-chart-monitor
redux-devtools-extension
redux-devtools-instrument
redux-starter-kit
redux-thunk
remotedev-app
reselect

Reset

user.avatar\_url



## Redux

user.name

user.bio

**BIO:** Redux is a predictable state container for JavaScript apps.

**LOCATION:** The Moon

user.location

**REPOSITORIES:**

d3-state-visualizer

react-redux

react-redux-benchmarks

redux

redux-devtools

redux-devtools-chart-monitor

repo.name

redux-devtools-extension

redux-devtools-instrument

redux-starter-kit

redux-thunk

remotedev-app

reselect

repos

Reset



Kao što je prikazano na prethodnoj slici, potrebno je prikazati sljedeće informacije o GitHub korisniku:

- Profil slika korisnika
- Ime korisnika
- Lokacija korisnika
- Detalji (bio) korisnika
- Lista koja sadrži imena korisnikovih repozitorija

Na detaljima korisnika, možemo na dno dodati button “Reset”. Klikom na taj button, brišemo trenutno stanje aplikacije i ponovno prikazujemo formu za unos korisničkog imena.

### Očekivanja:

- Kreirati novu aplikaciju pomoću create-react-app naredbe
- Napraviti više od jedne React komponente
  - barem jedna komponenta definirana funkcijom
  - barem jedna komponenta definirana klasom
- Koristiti props za komunikaciju među komponentama
  - definirati propTypes za sve komponente koje primaju propse
- Složiti komponente u parent-child odnosu
  - ispravno exportati child komponentu
  - ispravno importati child komponentu unutar parent komponente
- Koristiti state za čuvanje stanja aplikacije
  - čuvati stanje aplikacije u jednoj *Statefull* komponenti. Ova komponenta je “glavna” komponenta, u njoj čuvamo unešeno korisničko ime, korisnika i repozitorije dohvaćene s API-ja. Ova komponenta ima nekoliko child komponenti kojima kroz propse proslijeđuje vlastito stanje (GitHub korisnika i repozitorije)
- Koristiti listu i renderirati listu unutar JSX-a (lista repozitorija)
- Koristiti formu za unos GitHub korisničkog imena
- Koristiti događaje (evente)
  - npr. onChange prilikom unošenja GitHub korisničkog imena
  -



- npr. onClick za reakciju na klik buttona u formi za unos korisničkog imena

### Extra credit:

Promijeniti aplikaciju na način da se za upravljanje stanja koristi biblioteka Redux

- Napomena: prije implementacije ovog koraka, sačuvati trenutno programsko rješenje (aplikaciju koja čuva stanje unutar *statefull* komponente). Kopirajte kod u novi direktorij te dodatne promjene nastavite raditi u novom direktoriju.