



A Dive Into React

React je deklarativan, učinkovit i fleksibilan JavaScript library za izradu interaktivnih i složenih korisničkih sučelja

Sadržaj

1. Intro
2. React
 - a. Povijest
 - b. Osnovno o Reactu
 - c. Virtual DOM
 - d. JSX
 - e. Komponente
 - f. Hooks
3. Pluginovi
4. Redux
5. Create React App (CRA)
6. Reference



O MENI

Tomislav Leljak

20+ godina iskustva u IT
COO i Lead Frontend Developer @ AM2 Studio
Unapređenje procesa načina rada (workflow) je moja strast

<https://www.linkedin.com/in/leljak/>



REACT





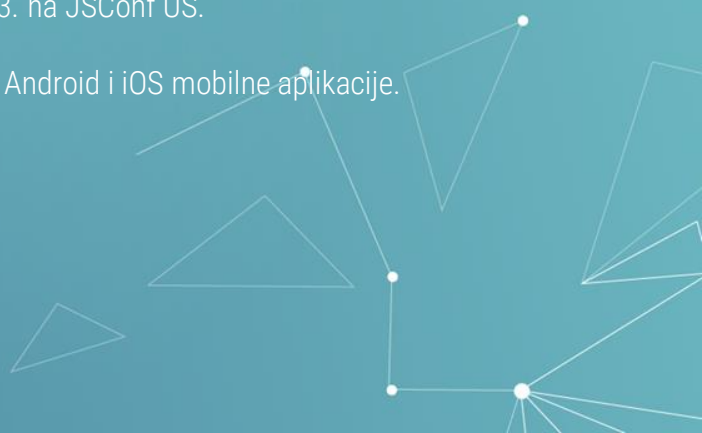
Povijest

Napravio ga je Jordan Walke, softverski inženjer u Facebooku, 2011. godine, a napravio ga je na uzoru XHP koji je jednostavan HTML komponentni framework za PHP.

Prvi slučaj upotrebe Reacta bio je na Facebookovom newsfeedu 2011., kasnije ga je Instagram iskoristio u svom sustavu.

S vremenom je React rastao, a Facebook je odlučio open sourceati ga u svibnju 2013. na JSConf US.

2015. godine na JSConf najavljen je i React Native koji je omogućio development za Android i iOS mobilne aplikacije.





Što je React

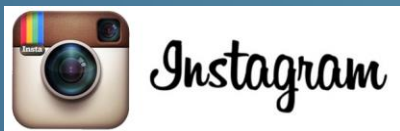
React je deklarativan, učinkovit i fleksibilan JavaScript library za izradu interaktivnih i složenih korisničkih sučelja, koji je izvorno razvio i vodi tim iz Facebooka.

React je view layer library, nije framework kao Backbone, Angular itd.; uz mnoštvo dodataka (pluginova) razvijenih u dev zajednici, React može postati punokrvni JavaScript framework

Bazira se na komponentama i koristi koncept zvan Virtual DOM



Tko koristi React?



facebook

NETFLIX

ATLASSIAN



I mnogi drugi...



Standardni development setup

Node i npm ili yarn

- Node.js je open-source, cross-platform, JavaScript runtime environment koji izvršava JavaScript kod izvan web preglednika
- npm - node package manager koji se sastoji od CLI i online repozitorija paketa/pluginova
- yarn - package manager

Babel - je open-source JavaScript transcompiler koji pretvara novije verzije JavaScript koda u starije verzije zbog kompatibilnosti

Webpack - je open-source JavaScript module bundler

ESLint (Airbnb) - je alat za analizu koda za prepoznavanje problematičnih uzoraka u kodu






Virtual DOM

DOM je sučelje za HTML i XML dokumente, definira logičku strukturu dokumenata i način kako se dokumentu pristupa i manipulira.

Virtual DOM (VDOM) je programski koncept u kojem idealni, odnosno "virtualni", prikaz sučelja ostaje u memoriji i sinkronizira se sa "stvarnim" DOM-om

DOM je spor
JavaScript je brz

Korištenje virtualnih DOM objekata omogućuje brze serije ažuriranja na stvarni DOM, uz veliku povećanu produktivnost u odnosu na česta kaskadna ažuriranja DOM stabla



JSX

JSX je XML proširenje sintakse na JavaScript bez ikakve definirane semantike.

Nije namijenjen da se implemetira u web preglednike već da ga predprocesori ili transpileri (Babel) pretvore standardni JavaScript

Umjesto da umjetno odvajate tehnologije stavljanjem markiranja i logike u odvojene datoteke, React razdvaja brige sa slabo spojenim jedinicama koje se nazivaju „komponente“ koje sadrže i jedno i drugo.

Ova smiješna sintaksa nije ni string ni HTML

```
const element = <h1>Hello, world!</h1>;
```

Komponente

Komponente su (reusable) sastavni dijelovi bilo koje React aplikacije, a tipična React aplikacija imat će ih puno.

Jednostavno rečeno, komponenta je JavaScript klasa ili funkcija koja opcionalno prihvaća ulaze tj. properties (props) i vraća React element koji opisuje dio korisničkog sučelja (UI)

Komponente mogu biti pisane kao ES6 klasa ili kao funkcija

```
const name = 'Ivan Horvat';
const element = <h1>Pozdrav, {name}</h1>;

ReactDOM.render(
  element,
  document.getElementById('root')
);
```



Komponente

Props - prenose se na komponentu s nadređene (parent) komponente i predstavljaju ulazne podatke za komponentu. Oni su nepromijenjivi iz same komponente

State - predstavlja unutarnje stanje komponente i promijenjivo je unutar same komponente

Kada se podaci o stanju komponente promijene, propsi ili state, prikazana markacija će se ažurirati ponovnim pozivanjem metode `render()`

Svaka komponenta Reacta ima životni ciklus (lifecycle methods) kojim možete nadzirati i manipulirati tijekom tri glavne faze. Tri faze su: **Mounting**, **Updating** i **Unmounting**

Napomena: state i lifecycle metode mogu se koristiti samo u komponentama pisanim kao klase.






Hooks

Hooks su novi dodatak u React 16.8., a omogućuju korištenje state i lifecycle metoda u komponentama pisanim kao funkcije

Najkorišteniji hooksi:

- `useState()`
 - `useEffect()`
-
- 

PLUGINOVİ



Najpopularniji pluginovi

- React Router
 - Redux Form
 - React Intl
 - Axios/Apollo
 - React-Redux
 - Redux Saga/Thunk (middleware)
 - Razni pluginovi za korisnička sučelja kao MaterialUI
 - Next.js
-

REDUX





React Redux

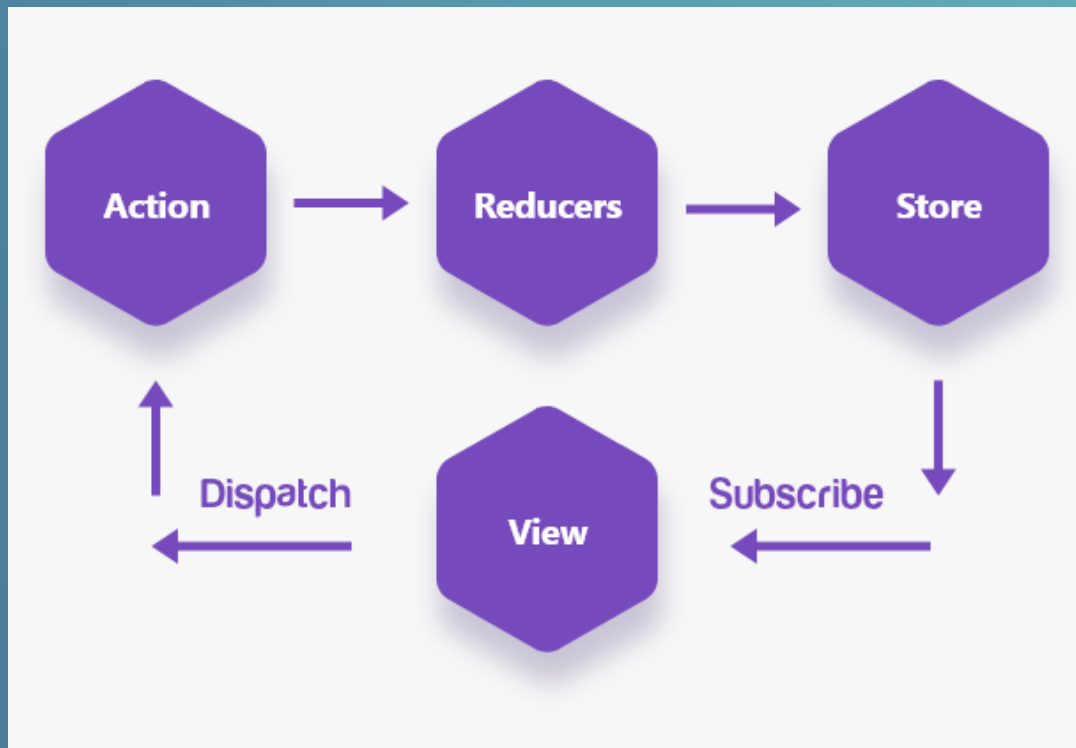
Redux se koristi za application level state

Omogućuje komponentama da čitaju podatke, pomoću selectora, iz Redux storea i šalju (dispatch) akcije u reducer radi ažuriranja podataka u Redux storeu

Application state je JSON objekt



Redux





CRA

Create React App




Create React App

Create React App je alat (koji su razvili developeri iz Facebooka) koji vam daje masivan početak prilikom izrade React aplikacija.

Štedi vas od dugotrajnog postavljanja i konfiguracije, jednostavno pokrenete jednu naredbu i Create React App postavi alate potrebne za pokretanje vašeg React projekta.

```
npx create-react-app my-app  
cd my-app  
npm start
```



Create React App

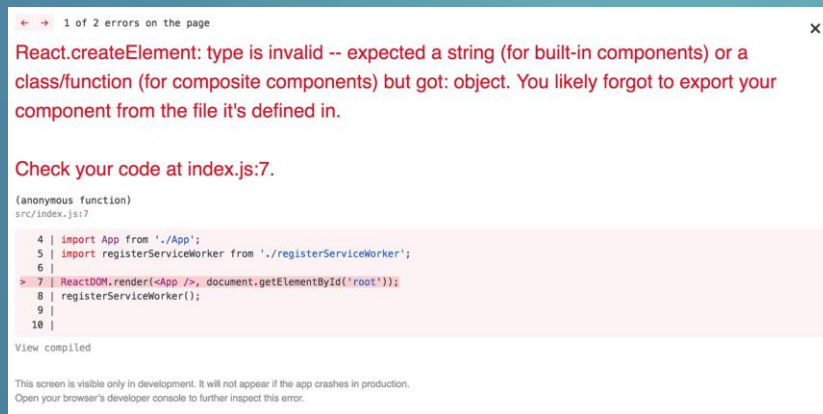
Babel, Webpack, ESLint

Skripte (taskove):

- start: development okruženje i server s hot reloading
- test: Built-in Jest unit testing
- build: Production Build, napravi minificirani bundle
- eject

Default Progressive Web Apps

Runtime error poruke u web pregledniku u development modu



The background of the slide is a solid teal color. Overlaid on this background is a complex, abstract geometric pattern. This pattern consists of numerous small white dots (nodes) connected by thin white lines (edges). These connections form a series of overlapping, irregular polygons of various sizes, creating a network-like or crystalline structure that is more dense in the upper right and lower right areas, and more sparse towards the left.

A Bonus

Storybook Demo



Reference

<https://nodejs.org/en/>

<https://www.npmjs.com/>

<https://reactjs.org/docs/getting-started.html>

<https://reacttraining.com/react-router/>

<https://react-redux.js.org/>

<https://create-react-app.dev/>

<https://webpack.js.org/>

<https://babeljs.io/>

<https://storybook.js.org/>



The background is a solid blue gradient. In the top-left corner, there is a complex network of white lines connecting several white dots of varying sizes. In the bottom-right corner, there are several white-outlined triangles of different sizes, some of which are connected by lines to a central point. Two thin white horizontal lines are also present: one in the upper right quadrant and one in the lower left quadrant.

HVALA!