JUNIORDev DIGITALNA DALMACIJA

Koponente sa stanjem > Virtualni DOM

Virtualni DOM

Virtualni DOM je važan koncept u Reactu i vezen je uz način na koji React prikazuje elemente na zaslonu preglednika i na koji način se prikaz osvježava nakon što se određeni elementi promijene.

Stvarni DOM

Prije nego prijeđemo na virtualni DOM, podsjetimo se kratko što je "stvarni" DOM. DOM je kratica od "*Document Object Model*" i predstavlja način na koji preglednik cijelu HTML strukturu stranice (znači korisničko sučelje ili UI) prikazuje kao strukturu stabla u kojem je svaki element predstavljen kao čvor. Svaki put kada napravimo neku promjenu na našoj stranici, preglednik mijenja DOM te ponovno renderira sučelje aplikacije. Upravo je taj proces ponovnog renderiranja najsporiji dio te kod složenijih sučelja može doći do dugog vremena osvježavanja što naravno umanjuje korisničko iskustvo rada sa stranicom.

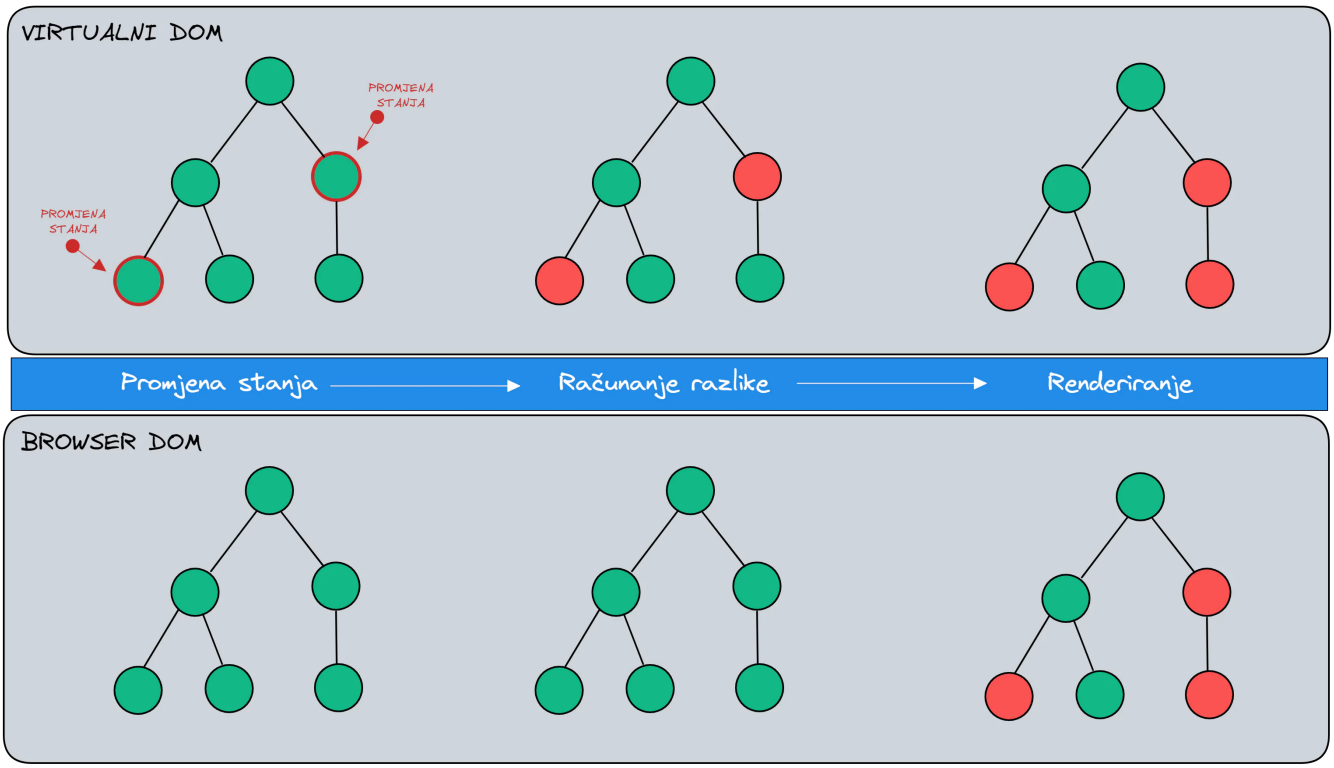
Virtualni DOM

React uvodi pojam virtualnog DOM-a kojeg možete zamisliti kao kopiju originalnog DOM-a. Svaki put kada napravimo promjenu u našoj aplikaciji, zapravo mijenjamo samo stanje virtualnog DOM-a, a ne stvarnog. Nakon što React detektira promjenu na virtualnom DOM-u usporediti će ga sa stvarnim kako bi dobio razliku (*diff*) između dva DOM-a. React zatim optimalni način na koji te promjene uvesti u stvarni DOM. Iako na prvu ovaj proces izgleda složeniji u stvarnosti je brži od direktne (neoptimizirane) manipulacije stvarnog DOM-a. Postoje i alternativni web okviri koji odbacuju koncept virtualnog DOM-a ali oni također imaju posebno razvijene načine na koji se osvježava prikaz.

React i virtualni DOM

Svaki dio korisničkog sučelja u Reactu je predstavljen u obliku komponente i svaka komponenta ima svoje unutarnje stanje. React prati svoje komponente i osluškuje promjene stanja. Kada se stanje neke komponente promijeni, React osvježava virtualno DOM stablo. Nakon što je stablo osvježeno, React uspoređuje trenutno stanje virtualnog DOM-a sa prethodnim stanjem virtualnog DOM-a. Ovaj proces uspoređivanja se naziva *diffing*.

Nakon što je React saznao koji dijelovi virtualnog DOM-a su se promijenili, React će osvježiti samo te objekte u stvarnom DOM-u. Dobra stvar je što je to sve događa u pozadini, bez potrebe za intervencijom programera. Zadatak programera je samo osvježiti stanje (vrijednosti) komponente, a React biblioteka će se pobrinuti za tehnički dio na koji način će se to izvesti.



Grupno (*batch*) osvježavanje

Još jedna od prednosti Reacta je što koristi tzv *batch update* mehanizam koji također doprinosi

boljim performansama aplikacije. Radi se o pristupu u kojem se osvježavanje stanja ne radi uvijek trenutno, već u serijama. Ukoliko u jednom trenutku promijenimo dvije vrijednosti u našoj aplikaciji - ona se neće osvježiti dva puta (po jednom za svaku promjenu) već će obje promjene biti poslane zajedno i aplikacija će se osvježiti samo jednom. Primjer ovakvog ponašanja ćemo vidjeti na idućim primjerima.

React za upravljanje stanjima funkcijskih komponenti koristi metodu ***useState*** koja spada u kategoriju tzv. ***React hooks***. React *hooks* su posebne funkcije koje su vezane uz upravljanje stanjima i životnim ciklusom (*lifecycle*) React komponenti i o njima ćemo detaljnije pričati u idućoj lekciji dok ćemo se sada fokusirati isključivo na *useState*.

Last updated on March 13, 2023

‹ Prikaz komponenti

Komponente sa stanjem ›