JUNIORDev
DIGITALNA DALMACIJA

Prikaz komponenti

Za početak ćemo se kratko vratiti na način prikaza (renderiranja) React komponenti. Ovo nam je važno jer kada uvedemo pojam unutarnjeg stanja komponenti moramo znati kako i kada se React komponente iscrtavaju na ekranu preglednika.

Započinjemo sa praznim React predloškom kao i na prethodnoj radionici, možete iskoristiti prethodni predložak ili napraviti novi sa naredbom:

```
npm create vite@latest vjezba02
```

ili ako koristite "klasični" CRA onda je naredba:

```
npx create-react-app vjezba02
```

Ostatak postupka pokretanja predloška bi vam trebao biti poznat od prošlog puta.

Ponovno ćemo izbrisati sav višak iz *App* komponente:

App.jsx

```
import './App.css'
function App() {

  return (
    <div>
        <p>Hello React!</p>
    </div>
  )
}
```

```
export default App
```

Zatim ćemo se fokusirati na *main.jsx* datoteku i naredbu za renderiranje:

main.jsx

```
ReactDOM.createRoot(document.getElementById('root')).render(
<React.StrictMode>
    <App />
</React.StrictMode>,
)
```

Razdvojiti ćemo ovu naredbu u dva dijela - dohvat *root* elementa i poziv metode za prikaz (renderiranje):

main.jsx

```
const root = ReactDOM.createRoot(document.getElementById('root'))

root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

Prikaz vrijednosti

Sada ćemo prije renderiranja napraviti jednu varijablu i poslati je komponenti kao *props* te ćemo naravno istu i ispisati na ekran:

main.jsx

```
const root = ReactDOM.createRoot(document.getElementById('root'))
let broj = 5;
root.render(
  <React.StrictMode>
    <App broj={broj} />
  </React.StrictMode>,
```

```
)
```

App.jsx

```
import './App.css'
function App(props) {

  return (
    <div>
      <p>Vrijednost broja: {props.broj}</p>
    </div>
  )
}
export default App
```

Što će se dogoditi ako promijenimo vrijednost varijable koju smo poslali komponenti?

main.jsx

```
let broj = 5;
root.render(
  <React.StrictMode>
    <App broj={broj} />
  </React.StrictMode>,
)
broj = 7;
```

Vrijednost broja na stranici se nije promijenila, zašto? React komponente su nepromjenjive (*immutable*) - jednom kada prikažemo neku komponentu, ne možemo mijenjati njene atribute niti djecu. Za prikaz prethodne promjene, bilo bi potrebno ponovno pozvati *render* metodu kako bi se naša komponenta ponovno iscrtala, ovaj put sa novom vrijednošću varijable:

main.jsx

```
let broj = 5;
root.render(
  <React.StrictMode>
    <App broj={broj} />
  </React.StrictMode>,
)
broj = 7;
```

```
root.render(
  <React.StrictMode>
    <App broj={broj} />
  </React.StrictMode>,
)
```

Očito je da ovo nije baš elegantan (niti praktičan) način za osvježavanje vrijednosti unutar naše aplikacije. Kroz iduće primjere ćemo vidjeti kako se to u praksi radi - React posjeduje mehanizme koji automatski pozivaju metodu za renderiranje u određenim uvjetima. Ovo je prikazano samo da steknete dojam što se događa u pozadini i kako bi se upoznali sa konceptom renderiranja i nepromjenjivosti.

Prije nego pređemo na same načine kako osvježiti vrijednosti unutar komponente, upoznati ćemo se sa još jednim konceptom koji je važan za razumijevanje React-a, a to je koncept virtualnog DOM-a.

Last updated on March 7, 2023

‹ Koponente sa stanjem

Virtualni DOM ›