

REACT - OPTIMALIZACE VÝKONU

React interně používá několik technik pro minimalizování počtu drahých DOM operací potřebné pro aktualizaci UI. Pro spoustu aplikací vede použití Reactu k rychlému UI, bez provedení příliš velké práce k optimalizaci výkonu. Nicméně je tu několik cest, jak můžeme naši React aplikaci zrychlit.

POUŽITÍ PRODUKČNÍHO BUILDU

V development módu nám React přidává spoustu užitečných varování. Ty ale dělají React větším a pomalejším, takže pro produkci je určitě lepší použít produkční verze.

Pokud si nejsme jistí, jestli je náš build proces nastaven správně, tak si můžeme nainstalovat React Developer Tools pro Chrome. Pokud navštívíme webovou stránku s Reactem v produkčním módu, tak bude mít ikona rozšíření tmavé pozadí, pokud v development módu, tak bude mít červené pozadí.

VIRTUALIZOVÁNÍ DLOUHÝCH SEZNAMŮ

Pokud naše aplikace renderuje dlouhé seznamy dat (stovky nebo tisíce řádků), tak je doporučeno použít techniku známou jako "windowing". Tato technika renderuje jen malou podmnožinu našich řádků a může dramaticky snížit čas, který se zabere renderováním komponenty a také počtu vytvořených nodes.

Knihovny react-window a react-virtualized jsou populární windowing knihovny. Poskytují několik znovupoužitelných komponent pro zobrazení listů, gridů a tabulkových dat.

POUŽITÍ METODY `shouldComponentUpdate`

Metodu `shouldComponentUpdate` můžeme u class komponenty implementovat ke kontrolování, jestli se má komponenta aktualizovat nebo ne.

```
shouldComponentUpdate(nextProps, nextState)
{
  if (
    this.props.color !== nextProps.color
  ) {
    return true;
  }
  if (
    this.state.count !== nextState.count
  ) {
    return true;
  }
  return false;
}
```

Ve většině případů ale můžeme dědit od `React.PureComponent`. Je to stejné jako bychom měli metodu `shouldComponentUpdate` s porovnáváním minulého a aktuálního stavu a props.

```
class CounterButton extends React.PureComponent
```