

REACT - ŘÍZENÍ EVENTŮ



Řízení eventů v Reactu je podobné jako řízení eventů u klasických DOM elementů. Je zde ale pár odlišností v syntaxi.

PŘIDÁNÍ EVENT LISTENERU

Event listenery se v reactu přidávají pomocí event atributů. Tyto atributy jsou v JSX psány camelCasem a můžeme jim nastavit jako hodnotu nějakou metodu komponenty. U class komponent je potřeba metody používané pro zpracovávání eventů bindovat.

```
class LoggingButton extends React.Component {
  constructor(props) {
    super(props);

    // když metodu handleClick předáváme, aby se později zavolala, tak v ní nebude klíčové slovo
    // this odkazovat na komponentu - tímto řádkem se to vyřeší (vytvoří se nová funkce, ve které
    // bude klíčové slovo this vždy odkazovat na komponentu)
    // - u function komponent tento problém řešit nemusíme, ale u class komponent ano
    this.handleClick = this.handleClick.bind(this);

    // - další možnost by byla vytvářet v render metodě namísto bindování novou funkci, to ale
    // není úplně ideální, protože by se tato funkce vytvářela při každém renderování
  }

  handleClick() {
    console.log("tlačítko bylo kliknuto");
  }

  render() {
    // atributem onClick se předává, která funkce se má po kliknutí na element zavolat
    return (
      <button onClick={this.handleClick}>
        Click me
      </button>
    );
  }
}
```

SYNTHETIC EVENT

V Reactu se do funkcí, které zpracovávají eventy, předává instance SyntheticEventu namísto nativního eventu. Má stejné rozhraní jako nativní event a obsahuje metody jako stopPropagation nebo preventDefault. Rozdíl je, že eventy fungují identicky ve všech prohlížečích.

```
function handleSubmit(e) {
  // SyntheticEvent má stejné rozhraní jako nativní Event
  e.preventDefault();
}
```