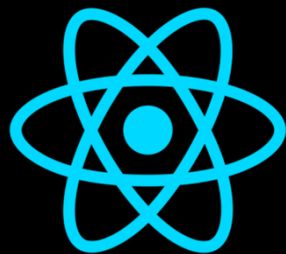


Szczecin on
Technologies

PODSTAWY REACT.JS



Cześć



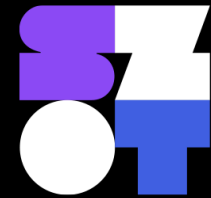
Paweł Skwirowski

Junior Front-end Developer w Tip and Donation

Znajdziecie mnie na GitHub'ie <https://github.com/skwirowski>

Agenda

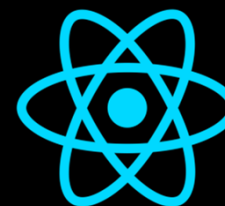
czyli co my tutaj zrobimy



- Czym jest React.js
- Czym jest Single Page Application
- Historia powstania React.js
- Czemu React.js? Charakterystyka biblioteki
- Ćwiczenia praktyczne

React.js

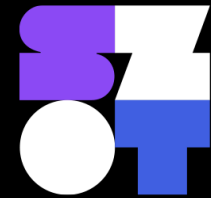
Czym jest?



- biblioteka open-source języka programowania JavaScript
- wykorzystywana jest do tworzenia interfejsów graficznych aplikacji internetowych
- pozwala w łatwy sposób zbudować aplikację SPA
- nie jest framework'iem JavaScript

Single Page Application

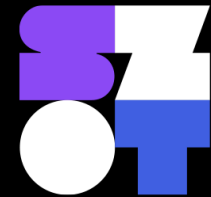
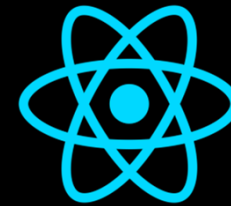
Czym jest?



- SPA to technologia, która pozwala wyświetlać poszczególne elementy strony, bez potrzeby ponownego załadowania całej strony

React.js

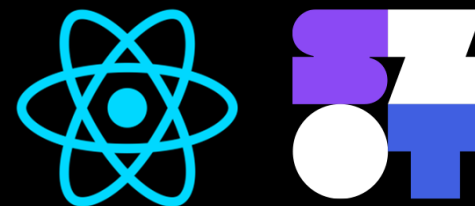
Jak powstał?



- W 2011 roku developerzy Facebook'a zaczęli mieć poważne problemy z utrzymaniem kodu.
- Wraz ze zwiększającą się ilością feature'ów aplikacji Facebooka, zespół potrzebował coraz więcej ludzi, którzy czuwali nad jej bezbłędnym działaniem.

React.js

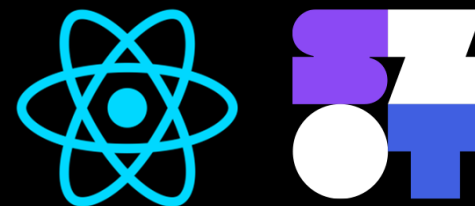
Jak powstał?



- Wciąż zwiększająca się liczba developerów oraz przybywających feature'ów spowolniła rozwój firmy.
- Z czasem aktualizacje zaczęły napływać kaskadowo, a ich aplikacja stawała się coraz trudniejsza w utrzymaniu.

React.js

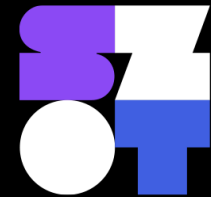
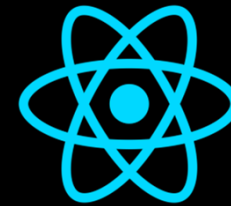
Jak powstał?



- Inżynierowie Facebook'a nie mogli nadążyć za aktualizacjami, dlatego kod wymagał pilnej poprawy i większej wydajności.
- Model aplikacji był poprawny ale potrzebowano zrobić coś z UX. Dlatego Jordan Walke stworzył prototyp, który był bardziej wydajny. Ten moment jest uznawany za narodziny React.js.

React.js

Jak powstał?



Jordan Walke

Brief React.js history timeline:

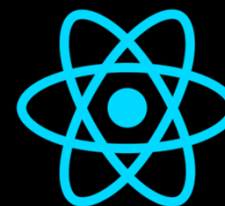
- 2010 - The first signs of React
- 2011 - An early prototype of React
- 2013 - The year of the Big Launch, React gets open sourced
- 2015 - React is Stable
- 2016 - React gets mainstream
- 2017 - The year of further improvements

<https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/>

<https://thenewstack.io/javascripts-history-and-how-it-led-to-reactjs/>

Czemu React.js?

Charakterystyka biblioteki

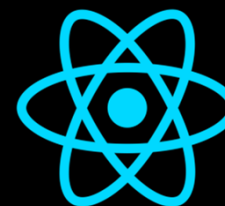


PROJEKTOWANIE OPARTE O KOMPONENTY

Komponenty są to funkcje i klasy. Przekazujemy im dane wejściowe i otrzymujemy dane wyjściowe w postaci interfejsu. Możemy tworzyć własne elementy HTML o funkcjonalności dostosowanej do naszych potrzeb.

Czemu React.js?

Charakterystyka biblioteki

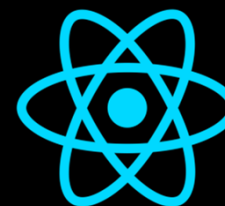


DEKLARATYWNOŚĆ

Deklaratywny styl pozwala kontrolować stan i “flow” aplikacji poprzez powiedzenie “to powinno wyglądać dokładnie tak”. Imperatywny styl działa odwrotnie, pozwala kontrolować stan aplikacji poprzez powiedzenie „trzeba wykonać następujące kroki”.

Czemu React.js?

Charakterystyka biblioteki

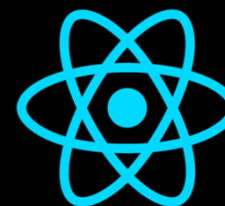


DEKLARATYWNOŚĆ

Podejście deklaratywne sprowadza się do tego, że nie musimy martwić się o całą implementację stanu aplikacji. Wystarczy określić zmiany jakie chcemy zaobserwować i odpowiednio zmodyfikować komponent. O odzwierciedlenie zmian w interfejsie zadba React bez naszej pomocy.

Czemu React.js?

Charakterystyka biblioteki

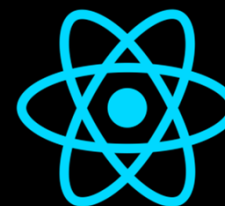


DEKLARATYWNOŚĆ

Deklaratywność Reacta zmniejsza liczbę błędów, zwiększa szybkość tworzenia aplikacji oraz jej wydajność.

Czemu React.js?

Charakterystyka biblioteki

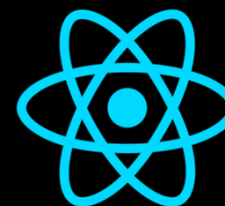


JEDNOKIERUNKOWY PRZEPŁYW DANYCH

W React zastosowany jest wzorzec jednokierunkowego przepływu danych w aplikacji. Dane zawsze płyną od komponentu-rodzica do zagnieżdżonych w nim komponentów-dzieci.

Czemu React.js?

Charakterystyka biblioteki



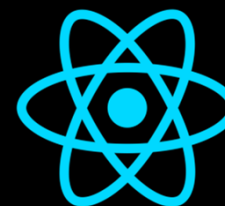
JEDNOKIERUNKOWY PRZEPŁYW DANYCH

Dzieje się to za pośrednictwem obiektu props. Jest on przeznaczony tylko do odczytu.

Komponent-dziecko może „poinformować” rodzica o swoim stanie poprzez przekazywane w props referencje do metod obsługujących zdarzenia w komponencie rodzicu.

Czemu React.js?

Charakterystyka biblioteki

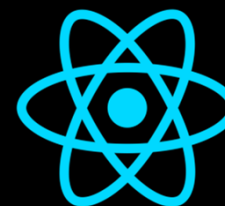


VIRTUAL DOM

Aktualizację drzewa DOM (Document Object Model) wspomaga Virtual DOM. Jest to technika, dzięki której renderowanie w przeglądarce jest dużo szybsze niż zwykle. Polega to na tym, że aktualizowane są tylko fragmenty drzewa, które uległy zmianie.

Czemu React.js?

Charakterystyka biblioteki



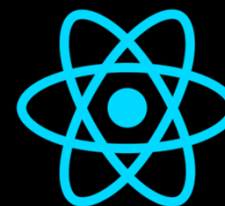
VIRTUAL DOM

Obiekt Virtual DOM jest reprezentacją obiektu DOM, ale w postaci lżejszej kopii.

Obiekt Virtual DOM ma takie same właściwości jak rzeczywisty obiekt DOM, ale nie posiada możliwości bezpośredniej zmiany tego co jest na ekranie.

Czemu React.js?

Charakterystyka biblioteki

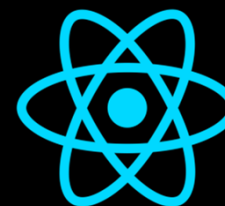


VIRTUAL DOM

Manipulowanie drzewem DOM jest powolne. Natomiast manipulacja obiektem Virtual DOM jest o wiele szybsza, ponieważ nie zmienia się nic na ekranie.

Czemu React.js?

Charakterystyka biblioteki



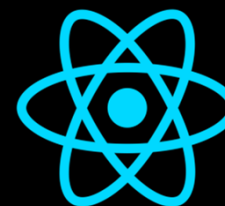
VIRTUAL DOM

W skrócie co się dzieje, kiedy próbujemy zaktualizować DOM w React:

- Całe wirtualne drzewo DOM zostaje zaktualizowane
- Virtual DOM zostaje porównany z tym jak wyglądał przed aktualizacją. React określa, które obiekty zostały zmienione.

Czemu React.js?

Charakterystyka biblioteki



VIRTUAL DOM

- Tylko zmienione obiekty zostają zaktualizowane w rzeczywistym drzewie DOM.
- Zmiany w rzeczywistym DOM'ie skutkują zmianą w widoku aplikacji.

Czemu React.js?

Charakterystyka biblioteki



W React.js zamiast regularnego JavaScript może zostać użyty JSX.

Czemu React.js?

Charakterystyka biblioteki



JSX

JSX (JavaScript XML) jest to rozszerzenie składni JavaScript pozwalające używać tagów HTML. Składnia ta jest przetwarzana przez odpowiednio skonfigurowany transpilator Babel.

Czemu React.js?

Charakterystyka biblioteki

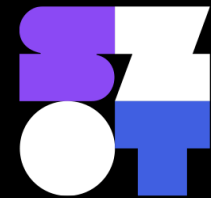


```
class Hello extends React.Component {  
  render() {  
    return <div>Hello {this.props.toWhat}</div>;  
  }  
}  
  
ReactDOM.render(  
  <Hello toWhat="World" />,  
  document.getElementById('root')  
);
```

Kod napisany przy użyciu składni JSX

Czemu React.js?

Charakterystyka biblioteki



```
class Hello extends React.Component {  
  render() {  
    return React.createElement('div', null, `Hello ${this.props.toWhat}`);  
  }  
}  
  
ReactDOM.render(  
  React.createElement(Hello, {toWhat: 'World'}, null),  
  document.getElementById('root')  
);
```

Kod napisany bez użycia składni JSX (czysty JavaScript)

Czemu React.js?

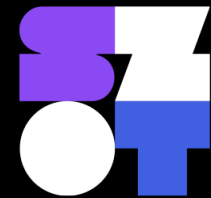
Charakterystyka biblioteki



Babel jest to kompilator JavaScript używany do konwertowania kodu ECMAScript 2015+ do wersji JavaScript wstecznie kompatybilnej z aktualnymi i starszymi przeglądarkami oraz środowiskami. Babel konwertuje również JSX do regularnego JavaScript.

Czemu React.js?

Charakterystyka biblioteki



```
// Babel Input: ES2015 arrow function  
[1, 2, 3].map((n) => n + 1);  
  
// Babel Output: ES5 equivalent  
[1, 2, 3].map(function(n) {  
  return n + 1;  
});
```

Przykład konwersji JavaScript ES6 do JavaScript ES5

Czemu React.js?

Charakterystyka biblioteki



webpack

Webpack to narzędzie służące m.in. do tworzenia bundli (pakietów, paczek) kodu z wielu różnych modułów. Wskazuje się jeden plik startowy, od którego zaczyna się analiza zależności i na tej podstawie powstaje drzewo zależności.

Czemu React.js?

Charakterystyka biblioteki



WEBPACK

Co oferuje Webpack:

- Składanie modułów w jeden plik
- Asynchroniczne ładowanie modułów
- Transpilowanie kodu ES6 do ES5
- Kompilacja kodu Sass/Less do CSS

Czemu React.js?

Charakterystyka biblioteki



WEBPACK

- Serwer testowy z automatycznym odświeżaniem przeglądarki
- Automatyczne wstawianie stylów i skryptów na stronę
- Tworzenie produkcyjnej wersji aplikacji

Czemu React.js?

Charakterystyka biblioteki



DUŻY EKOSYSTEM

- React cieszy się niesamowitą popularnością wśród programistów
- Wokół React'a powstała bardzo bogata społeczność
- Możemy szukać odpowiedzi na nurtujące nas pytania w jednym z kilkudziesięciu tysięcy wątków na stackoverflow.

Czemu React.js?

Charakterystyka biblioteki



DUŻY EKOSYSTEM

- Wiele narzędzi, które proponują coraz bardziej eleganckie rozwiązania dla najpopularniejszych problemów.
- Bogactwo materiałów edukacyjnych

Czemu React.js?

Charakterystyka biblioteki



WSZECHSTRONNOŚĆ

- Tworzenie aplikacji mobilnych przy użyciu React Native
- Kod React'a może być renderowany na serwerze za pomocą Next.js
- Tworzenie interfejsów użytkownika w 3D albo VR (Virtual Reality) przy użyciu React 360