

Aplikacje mobilne

Laboratorium 2 - Komponenty

Mateusz Pawełkiewicz

Wstęp:

Tworzenie aplikacji w React Native (RN) z użyciem komponentów jest standardowym podejściem, które pozwala na budowanie modularnych i łatwych w utrzymaniu aplikacji mobilnych. Oto kilka kroków i praktyk związanych z tym procesem:

1. Projektowanie komponentów:

- Najpierw należy zdefiniować, jakie komponenty będą potrzebne w Twojej aplikacji. Komponenty mogą być prostymi jednostkami interfejsu użytkownika, takimi jak przyciski, lub bardziej złożonymi kontenerami zawierającymi inne komponenty.

2. Tworzenie Komponentów:

- Zdefiniuj każdy komponent jako osobny plik w Twoim projekcie RN.

3. Stan i właściwości (Props):

- Zdefiniuj stan lokalny komponentu oraz właściwości, które komponent będzie przyjmować jako dane wejściowe.

- Użyj funkcji takich jak `useState` i `useEffect` do zarządzania stanem komponentu i reakcji na zmiany.

4. Komunikacja między komponentami:

- Przekaż dane między komponentami za pomocą właściwości.

Przykład (ToDoApp):

Link do przykładowej aplikacji:

<https://github.com/mpawelkiewiczpsk/toDoList/tree/main>

- Definiujemy prosty komponent, który będzie odpowiadał za wyświetlanie listy elementów przekazanych w propsach
- Dodajemy możliwość kliknięcia w dany element listy, aby oznaczyć, że dane zadanie zostało wykonane. W tym celu dodajmy metodę do aktualizacji stanu listy.
- Importujemy nasz komponent w głównym pliku App.js
- Zwracamy nasz komponent, gdzie w propsach podajemy listę oraz metodę, która będzie tę listę aktualizowała.

```
export default ToDoApp = ( { toDoList, changeStatus } ) => {  
  
  return(  
    <View style={styles.container}>  
  
      {toDoList.map(item => (  
        <TouchableOpacity key={item.id} onPress={() => changeStatus(item.id)}>  
          <View style={{...styles.entry, backgroundColor: item.done ? 'green': '#fff'}}>  
            <Text style={ styles.text }>{item.text}</Text>  
          </View>  
        </TouchableOpacity>  
      ))}  
    </View>  
  )  
  
}
```

Komponent wyświetlający listę

```
return (  
  <ToDoApp toDoList={list} changeStatus={changeStatus} />  
);
```

Wyświetlenie komponentu w głównym pliku App.js