

# Aplikacje mobilne

Laboratorium 5 – expo-sensors, camera

**Mateusz Pawełkiewicz**

1) **Expo-sensors** to zestaw modułów zawartych w Expo SDK, które umożliwiają dostęp do sensorów sprzętowych urządzeń mobilnych. Dzięki nim, deweloperzy aplikacji mobilnych mogą wykorzystywać dane z różnych sensorów, takich jak akcelerometr, żyroskop, magnetometr czy czujnik światła. Oto kilka kluczowych cech expo-sensors:

- Akcelerometr: Pozwala na śledzenie zmian w ruchu urządzenia wzdłuż osi X, Y i Z. Przydatny do wykrywania wstrząsów czy przechyłów urządzenia.

- Żyroskop: Mierzy prędkość obrotową urządzenia, co pozwala na wykrywanie obracania czy obracania się wokół osi.

- Magnetometr: Dostarcza informacji o polu magnetycznym wokół urządzenia. Może być używany do tworzenia kompasu cyfrowego.

- Barometr: Mierzy ciśnienie atmosferyczne. Może być wykorzystany do estymacji wysokości, na której znajduje się urządzenie.

- Czujnik światła: Wykrywa poziom światła w otoczeniu urządzenia. Użyteczny do dostosowywania jasności ekranu do warunków oświetleniowych.

- Podometr: Liczy kroki wykonane przez użytkownika, co jest przydatne w aplikacjach fitness i zdrowotnych.

2) **Expo Camera** to moduł w ramach Expo SDK dla React Native, który umożliwia interakcję z aparatem urządzenia mobilnego. Dzięki temu modułowi, deweloperzy mogą łatwo dodawać funkcjonalności związane z aparatem do swoich aplikacji React Native, takie jak robienie zdjęć czy nagrywanie filmów.

Kluczowe cechy **Expo Camera**:

1. **Proste API**: Moduł oferuje łatwe w użyciu API do zarządzania aparatem.
2. **Wsparcie dla różnych typów aparatów**: Obsługuje zarówno przedni, jak i tylny aparat.
3. **Kontrola nad parametrami aparatu**: Możliwość dostosowania różnych ustawień, takich jak zoom, ostrość, balans bieli.
4. **Zdjęcia i nagrania wideo**: Funkcje do robienia zdjęć i nagrywania filmów.

## Instalacja

```
npm install expo-camera
```

## Przykład użycia

```
import { Camera, CameraType } from 'expo-camera';
import { useState } from 'react';
import { Button, StyleSheet, Text, TouchableOpacity, View } from 'react-native';

export default function App() {
  const [type, setType] = useState(CameraType.back);
  const [permission, requestPermission] = Camera.useCameraPermissions();

  if (!permission) {
    // Camera permissions are still loading
    return <View />;
  }

  if (!permission.granted) {
    // Camera permissions are not granted yet
    return (
      <View style={styles.container}>
        <Text style={{ textAlign: 'center' }}>We need your permission to show the camera</Text>
        <Button onPress={requestPermission} title="grant permission" />
      </View>
    );
  }

  function toggleCameraType() {
    setType(current => (current === CameraType.back ? CameraType.front : CameraType.back));
  }

  return (
    <View style={styles.container}>
      <Camera style={styles.camera} type={type}>
        <View style={styles.buttonContainer}>
          <TouchableOpacity style={styles.button} onPress={toggleCameraType}>
            <Text style={styles.text}>Flip Camera</Text>
          </TouchableOpacity>
        </View>
      </Camera>
    </View>
  );
}
```

3) **Expo File System** jest modułem w ekosystemie Expo, który umożliwia interakcję z systemem plików na urządzeniu. Pozwala na wykonywanie różnorodnych operacji na plikach, takich jak odczyt, zapis, usuwanie plików i katalogów, a także pobieranie i przesyłanie plików.

#### Kluczowe Funkcje:

- a) **Odczyt i zapis plików:** Możesz odczytywać pliki z dysku, zapisywać do nich dane oraz tworzyć nowe pliki.
- b) **Pobieranie i przesyłanie:** Umożliwia pobieranie plików z Internetu na lokalne urządzenie oraz przesyłanie plików z urządzenia na serwer.
- c) **Zarządzanie plikami i katalogami:** Pozwala na tworzenie, usuwanie, przenoszenie, a także listowanie plików i katalogów.

```
import * as FileSystem from 'expo-file-system';

const filePath = FileSystem.documentDirectory + "example.txt";

// Zapisywanie tekstu do pliku
const saveFile = async () => {
  await FileSystem.writeAsStringAsync(filePath, "Przykładowy tekst");
};

// Odczytywanie tekstu z pliku
const readfile = async () => {
  const fileContent = await FileSystem.readAsStringAsync(filePath);
  console.log(fileContent);
};

saveFile();
readfile();
```

W tym przykładzie, funkcja **saveFile** zapisuje tekst "Przykładowy tekst" do pliku **example.txt** znajdującego się w katalogu dokumentów aplikacji. Funkcja **readFile** następnie odczytuje zawartość tego pliku i wyświetla ją w konsoli.