

# Fullstack Development

# Mobile development: React Native Services

| Camera | Location | Sensors

# Case study

- **Your boss:** *I need you to make a mirror app for me right now.*
- **You:** *Sure, give me .... minutes.*

# Initialize a project

- `npx create-expo-app -t expo-template-blank-typescript`
- `npx expo install expo-updates expo-camera`

./app.json

```
{
  "expo": {
    ...
    "plugins": [
      [
        "expo-camera",
        {
          "cameraPermission": "Allow $(PRODUCT_NAME) to access your camera."
        }
      ]
    ]
    ...
  }
}
```

## ./App.tsx

```
import { View, Button, SafeAreaView } from "react-native";
import { Camera, CameraType } from "expo-camera";
import { useState } from "react";
export default function App() {
  const [open, setOpen] = useState(false);
  const [cameraPermission, requestCameraPermission] =
    Camera.useCameraPermissions();

  // Camera is still loading
  if (!cameraPermission) return <View />;

  // Ask for permission
  if (!cameraPermission.granted) {
    return (
      <View style={{ flex: 1, justifyContent: "center" }}>
        <Button
          onPress={requestCameraPermission}
          title="We need your permission."
        />
      </View>
    );
  }
  return (
    <SafeAreaView style={{ flex: 1, paddingVertical: 40 }}>
      <Button onPress={() => setOpen((prev) => !prev)} title="Toggle" />
      {open ? (
        <Camera
          style={{ flex: 1 }}
          type={CameraType.front}
          onCameraReady={() => setOpen(true)}
        />
      ) : null}
    </SafeAreaView>
  );
}
```

# Build and deploy

- `eas init`
- `eas update:configure`
- `eas build:configure`
- `eas build --platform android --profile preview`

# Camera app

<https://github.com/fullstack-66/expo-camera>



# Nativewind

./utils/nativewind-styled.ts

```
import { View } from "react-native";  
import { styled } from "nativewind";  
  
export const StyledView = styled(View);
```

./app/index.tsx

```
<StyledView className="bg-purple-800 flex-1 items-center justify-center">  
  ...  
</StyledView>
```

# Some inconveniences

- Cannot wrap `styled` with `Button` component.
  - Does not have `style` attribute.
- Version conflict

```
{
  "dependencies": {
    "expo": "~49.0.8",
    "nativewind": "^2.0.11"
  },
  "devDependencies": {
    "tailwindcss": "^3.3.2" // Not 3.3.3
  }
}
```

# Camera mounting/unmounting problem

Only one Camera preview can be active at any given time. If you have multiple screens in your app, you should unmount Camera components whenever a screen is unfocused.

./app/camera/index.tsx

```
import { useIsFocused } from "@react-navigation/native";

export default function App() {
  // Use this to render camera when the screen is focused.
  const isFocused = useIsFocused();

  if (!isFocused) return <StyledView />;

  // Render camera here
  // return ...;
}
```

# Taking picture

- Get reference

```
const cameraRef = useRef<Camera>(null);  
  
return <Camera ref={cameraRef}>...</Camera>;
```

```
async function takePicture() {  
  if (!cameraRef.current) return;  
  let options = {  
    quality: 1,  
    base64: true,  
    exif: false,  
  };  
  
  try {  
    let newPhoto = await cameraRef.current.takePictureAsync(options);  
    setPicture(newPhoto);  
    router.push("/camera/review");  
  } catch (err) {  
    alert(JSON.stringify(err));  
  }  
}
```

The photo is actually stored locally.

# Saving photo

`./app/camera/review.tsx`

```
import * as MediaLibrary from "expo-media-library";

export default function Review() {
  const [mediaPermission, requestMediaPermission] =
    MediaLibrary.usePermissions();

  function savePhoto() {
    MediaLibrary.saveToLibraryAsync(picture!.uri).then(() => {
      setPicture(null);
      //...
    });
  }
  //...
}
```

# Location app

<https://github.com/fullstack-66/expo-location>



# SDK conflict

- Need to downgrade to SDK48

- `npx create-expo-app -t expo-template-blank-typescript@48`

```
{
  "dependencies": {
    "expo": "^48.0.0",
    "expo-location": "~15.1.1",
    "nativewind": "^2.0.11"
  },
  "devDependencies": {
    "tailwindcss": "^3.3.1" // Not 3.3.2 or 3.3.3
  }
}
```

```
import * as Location from "expo-location";

export default function App() {
  const [locationPermission, requestPermission] =
    Location.useForegroundPermissions();

  function getCurrentLocation() {
    Location.getCurrentPositionAsync({}).then((location) => {
      setLocation(location);
    });
  }
}
```

# Problem with Nativewind

- Nested flex box does not respect `gap` property.

```
function DisplayLocation({ location }: DisplayLocationProps) {
  if (!location) return <StyledView />;
  return (
    // I cannot set gap here. 🙅🙅🙅🙅🙅🙅🙅🙅🙅🙅
    <StyledView className="bg-gray-100 p-4 rounded-xl flex-col mb-6">
      <ListItem title="Latitude" value={location.coords.latitude} />
      <ListItem title="Longitude" value={location.coords.longitude} />
      <ListItem title="Accuracy" value={location.coords.accuracy} />
      <ListItem title="Altitude" value={location.coords.altitude} />
      <ListItem title="Heading" value={location.coords.heading} />
      <ListItem title="Speed" value={location.coords.speed} />
      <ListItem title="Timestamp" value={location.timestamp} />
    </StyledView>
  );
}
```

## Other locations-relate things to do.

- Background fetching
- Geofencing
- Map

# Sensor app

<https://github.com/fullstack-66/expo-sensors>

```
import { Accelerometer, AccelerometerMeasurement } from "expo-sensors";

export default function App() {
  const _subscribe = () => {
    setSubscription(Accelerometer.addListener(setData));
  };

  const _unsubscribe = () => {
    subscription && subscription.remove();
    Accelerometer.removeAllListeners();
    setSubscription(null);
  };
}
```

# Update interval

```
const _slow = () => Accelerometer.setUpdateInterval(1000);  
const _fast = () => Accelerometer.setUpdateInterval(500);
```

# Other sensors

```
import {  
  Accelerometer,  
  Barometer,  
  DeviceMotion,  
  Gyroscope,  
  LightSensor,  
  Magnetometer,  
  MagnetometerUncalibrated,  
  Pedometer,  
} from "expo-sensors";
```