Extending IoT to Mobile Phones

Chunxu Yang

Why mobile phones?

- Easy to get
- Easy to manipulate
- Easy to visualize

How to make an App?

Native	Frameworks	Web
Very difficult	Need some effort	Easy for web developers
Best performance	Medium performance	Low performance
Rich APIs	Relatively rich APIs	Not many APIs, especially in IoT apps
Swift / Java	JavaScript / Dart	HTML / JavaScript / CSS

Why use frameworks?

- Quick prototype
- Extensible APIs
- Multiple platforms

Which framework to use?

Flutter

- Use **Dart** to develop
- Better performance
- Cooperate with **Figma**

React Native

- Simple **React** syntax with **JavaScript**
- Easy to start
- Quickly debug and push with **Expo**

Why React Native with Expo?

- We are using JavaScript already, why Dart?
- Really great Docs and active community
- Good performance for small projects
- Simple to test and deploy

But...

- Some native modules don't support expo...
    - Eject the project?
    - Or just find alternatives...
    - Or just talk with the PM
- I don't understand Mobile Phone at all...
    - Don't worry! React Native and Expo is what you need
    - Go to the docs

Dive in Expo

- Precondition:
    - Node.js
    - NPM
- Installation
    - `npm i -g expo-cli`
    - `npx create-expo-app $YOUR_APP_NAME`
    - `cd $YOUR_APP_NAME`

- Additional dependencies
    - Sensors - Expo Documentation
    - `npx expo install expo-sensors`
- Let's expo!
    - expo start
    - Download **Expo Go** from Expo
    - Scan the QRCode
        - In the same WiFi

Do the IoT things

Import library first:

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

Define Some functions:

```
const [accelerometerData, setAccelerometerData] = React.useState({
  x: 0,
  y: 0,
  z: 0,
});

const [subscription, setSubscription] = React.useState(null);

const _slow = () => {
  Accelerometer.setUpdateInterval(1000);
};

const _fast = () => {
  Accelerometer.setUpdateInterval(16);
};

const _subscribe = () => {
  setSubscription(
    Accelerometer.addListener((accelerometerData) => {
      setAccelerometerData(accelerometerData);
    })
  );
};

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

React.useEffect(() => {
  // set Accelerometer to update every 1000ms
  _fast();
  _subscribe();
  return () => _unsubscribe();
}, []);
```

Manage the component:

```
<View>
  <Text>X: {accelerometerData.x}</Text>
  <Text>X: {accelerometerData.y}</Text>
  <Text>X: {accelerometerData.z}</Text>
</View>
```

Maybe fancier?

```
<View
  style={[
    styles.dot,
    {
      backgroundColor: "red",
      top: 100 + accelerometerData.x * 100,
    },
  ]}
/>
<View
  style={[
    styles.dot,
    {
      backgroundColor: "green",
      top: 100 + accelerometerData.y * 100,
    },
  ]}
/>
<View
  style={[
    styles.dot,
    {
      backgroundColor: "blue",
      top: 100 + accelerometerData.z * 100,
    },
  ]}
/>
```

And more? Try your self!

- Some sensors: Sensors - Expo Documentation
- WiFi, ble, or more: Network - Expo Documentation
- Haptics: Haptics - Expo Documentation
- More...