アジェンダ

1. 開発環境を構築しよう (10:30-11:30)
2. Hello Worldを表示してみよう (11:30-11:45)
3. 画像を投稿してみよう (13:00-14:00)
4. 自分だけの画面を作成してみよう (14:00-15:00)
5. 発表会 (15:00-16:00)

概要

今回は、「Node.js」というサーバサイドでも動かすことができるフレームワークを使用します。 画面などを表示する方（フロントエンドと呼びます。）は、React Nativeというフレームワークを利用します。React Nativeは、Facebook等にも使われているフレームワークです。 また、今回はiOS/Androidアプリとういうことで「EXPO」というサービスを利用してアプリを開発していきます。 今回は、初めてということもあり、あまり難しいことはしません。わからないときは、遠慮なく質問してください。

1. 開発環境を構築しよう

1.1 nodebrewをインストールします。

1.1.1 とりあえずインストール

$curl -L git.io/nodebrew | perl - setup

1.1.2 .bash_profileの作成

$touch .bash_profile

1.1.3 .bash_profileに以下のパスを追加

export PATH=$HOME/.nodebrew/current/bin:$PATH

1.1.4 確認。以下のコマンドを実行

$nodebrew help

何か表示されればOKです。

1.2 Node.js をインストール

1.2.1 とりあえず、最新版をインストール

$nodebrew install v8.9.4

1.2.2 このバージョン使いますよ！と宣言しときます。

$nodebrew use v8.9.4

1.2.3 確認

$node -v

$npm -v

何か表示されればOKです。

1.2.4 今後の開発スピード上げるためにもう一息。もう一個インストール

$npm install -g yarn

1.3 EXPOで開発環境を構築 をインストール

1.3.1 3度目のとりあえずインストール

yarn global add create-react-native-app

1.3.2 プロジェクトの作成

(今回は、プロジェクト名は「KouenApp」にします。)

$create-react-native-app KouenApp

(入力を求められるので、「KouenApp」と入力。そのあと、yを押してEnter)

1.3.3 作業場所に移動

$cd KouenApp

1.3.4 動かしてみる

$yarn start

QRCodeが表示されればOKです。

1.4 iPhone/Androidにアプリをインストール

1.4.1 expo clientアプリをインストール

アプリインストール後にアプリから、1.3.4で表示されているQRCodeを読み込み、スマホで何か表示されればOKです。

2 . Hello Worldを表示してみよう

2.1 開発を行うためにCSCodeをインストールしよう（絶対ではないです。）

2.2 App.jsを編集します。

App.jsに記載されている「Open up App.js to start working on your app!」を「Hello World!」に変えて、スマホの画面をリロードしてください。
画面の文字が切り替わるはずです。

3 . 画像を投稿してみよう

概要

今回は、CloudサービスでGoogleが提供している、Firebaseというサービスを利用します。（無料です。）
他のサービスで有名なところでは、AWSやMicrosoftなどがクラウドサービスを行なっています。今回のような、簡易なアプリなどを作成するときは、無料のサービスでまずは作成してみるなどオススメです！
今回は、Firebaseのアカウントはこちらでよ用意した共通のアカウントを使います。

アカウント：c4f.hjm.murakami20191124@gmail.com パスワード：

3.1 画像投稿を行う画面を作成しよう。

少し作業が増えますが頑張りましょう。

3.1.1 一旦今起動しているアプリを停止します

ターミナルもしくは、コマンドプロンプトで　Ctrl+C

3.1.2 モジュールのインストールを行います

以下のコマンドを、ターミナルかコマンドプロンプトで行なってください。
($以下全てコマンド)

$yarn add react-native-elements
$yarn add react-navigation
$yarn add react-native-gesture-handler react-native-reanimated
$yarn add expo-image-picker
$yarn add firebase

3.1.3 ディレクトリの作成

componentsディレクトリの作成を行います。

App.jsがある階層に「components」という名前でディレクトリの作成をしてください。
作成の方法は何でも良いです。

3.1.4 遷移先の画面ファイルの作成

componentsディレクトリにファイルの作成を行います。２つファイルを作成します。
「ImageScreen.js」、「HomeScreen.js」という名前のファイルを作成します。

3.1.5 ImageScreen.jsの編集

以下のように編集してください。

```
import React from 'react'
import { Button, View, Text } from 'react-native'
import * as firebase from 'firebase';
import 'firebase/firestore';
import * as ImagePicker from 'expo-image-picker';
import Constants from 'expo-constants';
import * as Permissions from 'expo-permissions';


const firebaseConfig = {
  apiKey: "AIzaSyA8RoG2t_5JTqPiB3qfNk7Z85IuSkwYeGM",
  databaseURL:
"https://console.firebase.google.com/project/code4fukuokakouenapp/database
/firestore/data~2Ftest",
  projectId: "code4fukuokakouenapp",
  storageBucket: "gs://code4fukuokakouenapp.appspot.com",
  appId: "1:634290705297:ios:673baf10794af79c290030"
};

firebase.initializeApp(firebaseConfig);
const db = firebase.firestore();

export default class ImageScreen extends React.Component {

  componentDidMount = () => {
    this.getPermissionAsync();
  }

  getPermissionAsync = async () => {
    if (Constants.platform.ios) {
        const { status } = await
Permissions.askAsync(Permissions.CAMERA_ROLL);
        if (status !== 'granted') {
            alert('カメラ利用の許可が必要です。')
        }
    }
  }

  onChooseImagePress = async () => {
```

```
    let result = await ImagePicker.launchCameraAsync();
    //ライブラリから選ぶ場合
    // let result = await ImagePicker.launchImageLibraryAsyn();

    if (!result.cancelled) {
        this.uploadImage(result.uri, "test-image")
            .then(() => {
                alert("success");
            })
            .catch(e => {
                alert(JSON.stringify(e));
            })
    }
  }
  uploadImage = async (uri, imageName) => {

    //元のファイルからblobを生成
    const response = await fetch(uri);
    const blob = await response.blob();

    //firestoreの保存場所指定
    var ref = firebase.storage().ref().child("images/" + imageName);

    //保存場所にput
    return ref.put(blob);
  }


  render() {
    return (
      <View style={{ flex: 1, justifyContent: 'center', alignItems:
'center' }}>
          <Text>App</Text>
          <Button
              title="写真を選択"
              onPress={() => this.onChooseImagePress()}
          />
      </View>
  );
  }
}
```

3.1.6 Home.jsの編集

以下のように編集してください。

```
import React from 'react'
import { Button, View, Text } from 'react-native'

class HomeScreen extends React.Component {
```

```
  render(){
    return (
      <View >
        <Text>Code 4 Fukuoka</Text>
        <Text>公園アプリ</Text>
        <Button
          title="投稿する"
          onPress={() => this.props.navigation.navigate('Image')}
        />
      </View>
    );
  }
}

export default HomeScreen;
```

3.1.7 App.jsの編集

App.jsを以下のように編集してください。

```
import React from 'react';
import { createAppContainer } from 'react-navigation';
import { createStackNavigator } from 'react-navigation-stack';
import HomeScreen from './components/HomeScreen'
import ImageScreen from './components/ImageScreen'



const RootStack = createStackNavigator(
  {
    Home: HomeScreen,
    Details: DetailsScreen,
    Image:ImageScreen,
    Map:GoogleMapScreen
  },
  {
    initialRouteName: 'Home',
  }
);


const AppContainer = createAppContainer(RootStack);

export default class App extends React.Component {
  render() {
    return <AppContainer />;
  }
}
```

3.1.8 アプリを起動して、画面の確認

以下のコマンドを、ターミナルかコマンドプロンプトで行なってください。
($以下全てコマンド)

$yarn start

（スマホでQRCodeを読み込み）

画面が変わってたらOKです。
画面から「投稿する」をクリックすると画面遷移できます。
画面遷移後で、「写真を選択」クリックすると写真選択の画面が表示されます。

3.1.9 Home画面のデザインを変更してみよう

HomeScreen.jsを以下のように変更します。

```
import React from 'react'
import { Button, View, Text, StyleSheet } from 'react-native'

class HomeScreen extends React.Component {
  render(){
    return (
      <View >
        <Text>Code 4 Fukuoka</Text>
        <Text>公園アプリ</Text>
        <View style={styles.buttonContainer}>
        <Button
          title="投稿する"
          onPress={() => this.props.navigation.navigate('Picture')}
        />
        </View>
        <View style={styles.buttonContainer}>
        <Button
          title="公園一覧"
          onPress={() => this.props.navigation.navigate('List')}
        />
        </View>
      </View>
    );
  }
}


const styles = StyleSheet.create({
  buttonContainer: {
    height: 100,
    width: 200,
    padding: 10,
    backgroundColor: '#FFFFFF',
```

```
    margin: 3
  },
});

export default HomeScreen;
```

3.1.10 4章に向けての準備

App.jsがある場所に「Firebase.js」という名前でファイルを作成します。

Firebase.js

```
// Config file
import * as firebase from "firebase";

const firebaseConfig = {
    apiKey: "AIzaSyA8RoG2t_5JTqPiB3qfNk7Z85IuSkwYeGM",
    databaseURL: "https://code4fukuokakouenapp.firebaseio.com",
    projectId: "code4fukuokakouenapp",
    storageBucket: "gs://code4fukuokakouenapp.appspot.com",
    appId: "1:634290705297:ios:673baf10794af79c290030"
};


export default !firebase.apps.length ?
firebase.initializeApp(firebaseConfig) : firebase.app();
```

「PictureScrenn.js」ファイルを少し修正します。

変更前

```
import * as firebase from 'firebase';
import 'firebase/firestore';//変更箇所
import * as ImagePicker from 'expo-image-picker';
import Constants from 'expo-constants';
import * as Permissions from 'expo-permissions';

//削除
const firebaseConfig = {
  apiKey: "AIzaSyA8RoG2t_5JTqPiB3qfNk7Z85IuSkwYeGM",
  databaseURL:
"https://console.firebase.google.com/project/code4fukuokakouenapp/database
/firestore/data~2Ftest",
  projectId: "code4fukuokakouenapp",
  storageBucket: "gs://code4fukuokakouenapp.appspot.com",
  appId: "1:634290705297:ios:673baf10794af79c290030"
};
```

```
//


//削除
firebase.initializeApp(firebaseConfig);
//削除
const db = firebase.firestore();
```

変更内容

```
import firebase from '../Firebase';
```

４. 自分だけの画面を作成してみよう

FacebookやInstgramなどを参考に、画面を作成してみよう。