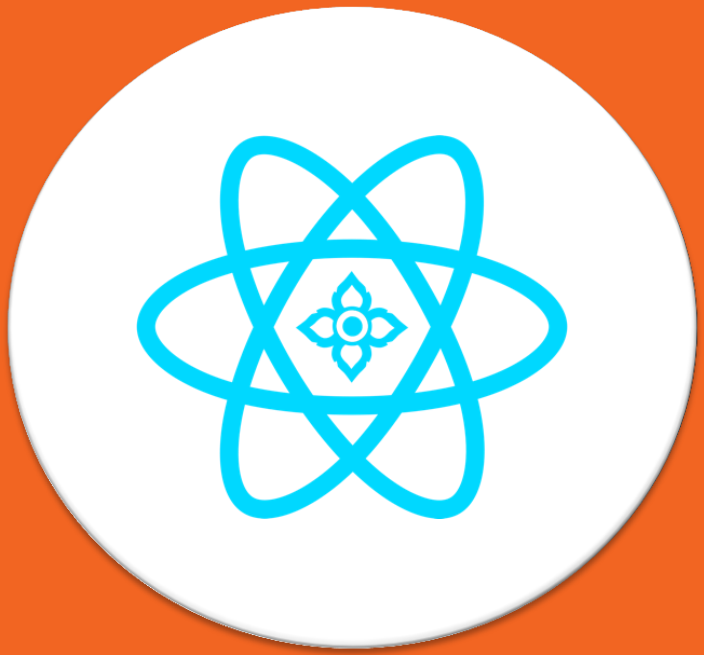
Fpt University

FPT POLYTECHNIC



LẬP TRÌNH ĐA NỀN TẢNG VỚI REACT

BÀI 7: COMPONENT VÀ API

PHẦN 1: SỬ DỤNG ALERT VÀ GEOLOCATION

www.poly.edu.vn

Mục tiêu

- Sử dụng **Alert** trong React Native
- Sử dụng **Geolocation** trong React Native
- Sử dụng **AsyncStorage** trong React Native
- Sử dụng **Camera** trong React Native

THÀNH PHẦN VÀ API: ALERT

- Alert: Thành phần này dùng để đưa ra thông báo cho người dùng
- Ví dụ: Ta sẽ tạo một custom alert như sau
- File **src/components/home/Home.js**

home.js

```
1  import React from 'react'
2  import AlertExample from './AlertExample.js'
3
4  const Home = () => {
5      return (
6          <AlertExample />
7      )
8  }
9  export default Home
```

THÀNH PHẦN VÀ API: ALERT

- Tiếp theo ta sẽ tạo một nút để kích hoạt chức năng showAlert.
- File **src/components/home/AlertExample.js**

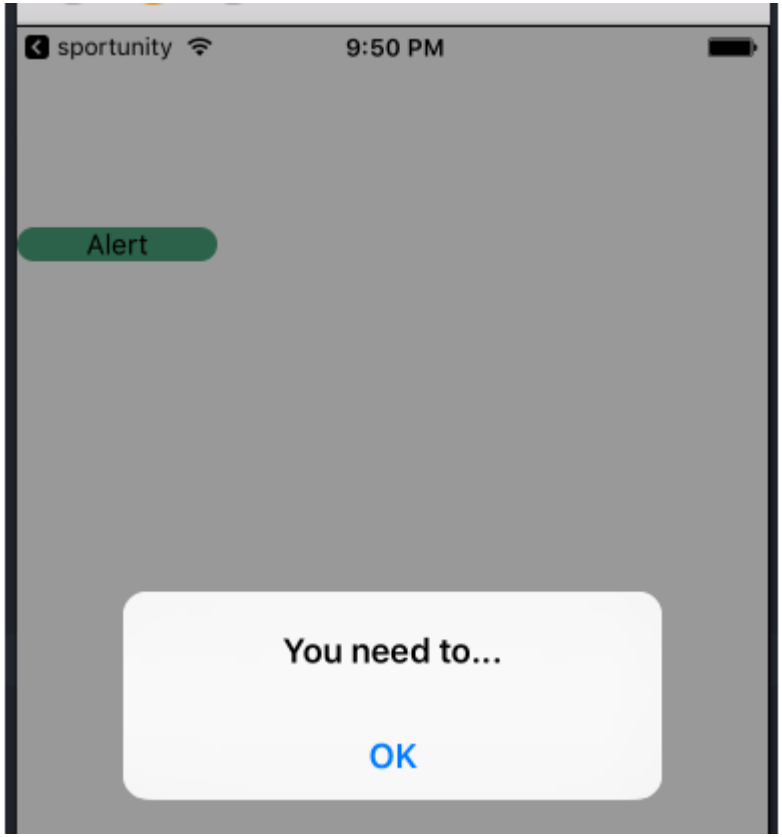
THÀNH PHẦN VÀ API: ALERT

- File **src/components/home/AlertExample.js**

```
1  import React from 'react'
2  import { Alert, Text, TouchableOpacity, StyleSheet } from 'react-native'
3
4  const AlertExample = () => {
5      const showAlert = () => {
6          Alert.alert(
7              'You need to...'
8          )
9      }
10     return (
11         <TouchableOpacity onPress = {showAlert} style = {styles.button}>
12             <Text>Alert</Text>
13         </TouchableOpacity>
14     )
15 }
16 export default AlertExample
17
18 const styles = StyleSheet.create ({
19     button: {
20         backgroundColor: '#4ba37b',
21         width: 100,
22         borderRadius: 50,
23         alignItems: 'center',
24         marginTop: 100
25     }
26 })
```

THÀNH PHẦN VÀ API: ALERT

- Chạy chương trình sẽ cho kết quả sau



THÀNH PHẦN VÀ API: GEOLOCATION

- **Geolocation**: dùng để xác định vị trí người dùng sử dụng smartphone
- Chúng ta sẽ bắt đầu bằng cách thiết lập trạng thái nhằm giữ ban đầu và vị trí cuối cùng.

THÀNH PHẦN VÀ API: GEOLOCATION

- Bây giờ ta cần phải có được vị trí hiện tại của thiết bị khi một thành phần được gắn kết bằng cách sử dụng **navigator.geolocation.getCurrentPosition**.

THÀNH PHẦN VÀ API: GEOLOCATION

- Chúng ta sẽ dùng React để cập nhật state.
- navigator.geolocation.watchPosition được sử dụng để theo dõi vị trí của người dùng.
- Bạn cần khai báo quyền sau trong AndroidManifest.xml

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
```

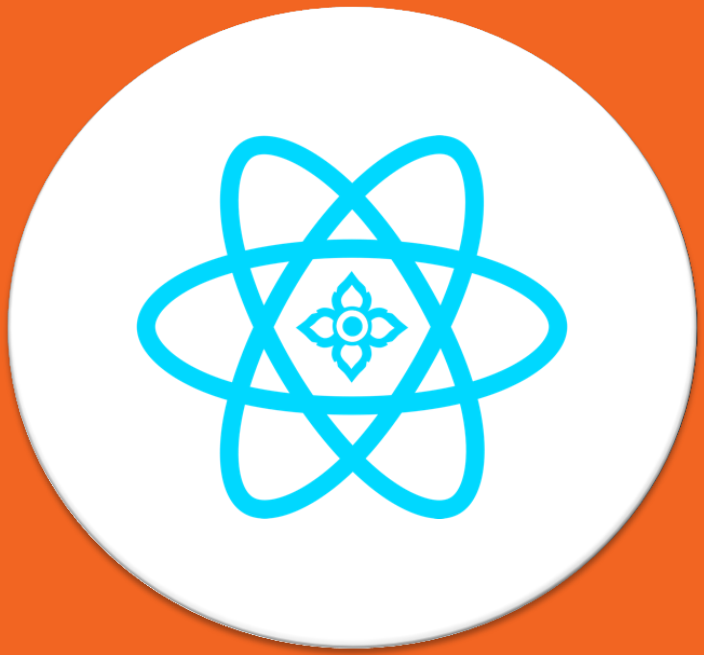
- Khuyến cáo nên sử dụng Android API >= 18

THÀNH PHẦN VÀ API: GEOLOCATION

- Khi muốn xóa vị trí lưu trữ, ta dùng phương thức **Geolocation.clearWatch()**
- **Geolocation** là một giao diện
- Lưu ý: với IOS bạn cần đưa key này vào info.plish
  - NSLocationWhenInUseUsageDescription
- Ý nghĩa của phương thức static **setRNConfiguration**(config)
  - Đặt các tùy chọn cấu hình để được sử dụng trong tất cả các yêu cầu vị trí.

Fpt University

FPT POLYTECHNIC



LẬP TRÌNH ĐA NỀN TẢNG VỚI REACT

BÀI 7: COMPONENT VÀ API

PHẦN 2: SỬ DỤNG ASYNCSTORAGE VÀ CAMERA

www.poly.edu.vn

THÀNH PHẦN VÀ API: ASYNCSTORAGE

- **AsyncStorage**: dùng để đồng bộ dữ liệu với bộ nhớ lưu trữ
- Ví dụ:
  - Name từ trạng thái ban đầu là chuỗi rỗng. Chúng ta sẽ cập nhật nó từ bộ nhớ lưu trữ liên tục khi thành phần được gắn kết.
  - setName sẽ lấy văn bản từ trường input và lưu nó bằng AsyncStorage và cập nhật trạng thái.

- File **src/components/home/Home.js**

home.js

```
1  import React from 'react'
2  import AsyncStorageExample from './AsyncStorageExample.js'
3
4  const Home = () => {
5      return (
6          <AsyncStorageExample />
7      )
8  }
9  export default Home
```

THÀNH PHẦN VÀ API: ASYNCSTORAGE

- File **src/components/home/AsyncStorageExample**

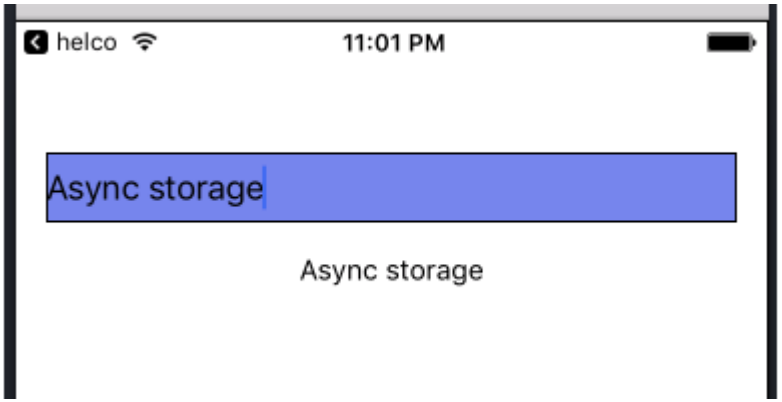
```
1  import React, { Component } from 'react'
2  import { AsyncStorage, Text, View, TextInput, StyleSheet }
3  from 'react-native'
4
5  class AsyncStorageExample extends Component {
6      state = {
7          'name': ''
8      }
9      componentDidMount = () ⇒ AsyncStorage.getItem('name').then((value)
10         ⇒ this.setState({ 'name': value }))
11
12     setName = (value) ⇒ {
13         AsyncStorage.setItem('name', value);
14         this.setState({ 'name': value });
15     }
16     render() {
17         return (
18             <View style = {styles.container}>
19                 <TextInput style = {styles.textInput} autoCapitalize = 'none'
20                     onChangeText = {this.setName}/>
21                 <Text>
22                     {this.state.name}
23                 </Text>
24             </View>
25         )
26     }
27 }
```

- File **src/components/home/AsyncStorageExample**

```
28  export default AsyncStorageExample
29
30  const styles = StyleSheet.create ({
31     container: {
32        flex: 1,
33        alignItems: 'center',
34        marginTop: 50
35     },
36     textInput: {
37        margin: 15,
38        height: 35,
39        borderWidth: 1,
40        backgroundColor: '#7685ed'
41     }
42  })
```

THÀNH PHẦN VÀ API: ASYNCSTORAGE

- Khi chạy ứng dụng, chúng ta có thể cập nhật văn bản bằng cách nhập vào trường input.



THÀNH PHẦN VÀ API: ASYNCSTORAGE

- Chú ý: Khi đồng bộ dữ liệu lên server nodejs, ta cần nhớ lại một số phương thức xử lý trên server như sau:
- Cần tạo tiến trình con để đồng bộ, tiến trình con đó là child process
- Phương thức exec() sẽ được sử dụng để chạy tiến trình con
- Ta cần truyền tham số timeout để xác định thời gian kết thúc đồng bộ
- Khi làm việc với tiến trình con để đồng bộ dữ liệu, ta nên dùng phương thức spawn()

THÀNH PHẦN VÀ API: ASYNCSTORAGE

- Chú ý: Khi đồng bộ dữ liệu lên server nodejs, ta cần nhớ lại một số phương thức xử lý trên server như sau:
- Khi đồng bộ không thành công, đôi khi ta phải kill tiến trình con. Trong trường hợp không kill được thì chương trình sẽ báo lỗi (error)

THÀNH PHẦN VÀ API: ASYNCSTORAGE

- Khi xử lý đồng bộ, nếu muốn thi hành trực tiếp trên file mà không phải shell, ta dùng phương thức execFile()
- Sự kiện close xuất hiện khi standard stream của tiến trình con đồng bộ dữ liệu đóng lại
- Sau khi đồng bộ dữ liệu xong bạn cần đóng kết nối bằng sự kiện disconnect

THÀNH PHẦN VÀ API: CAMERA

- **Camera**: dùng để quay clip, chụp ảnh trong React Native

TỔNG KẾT

- Sử dụng **Alert** trong React Native
- Sử dụng **Geolocation** trong React Native
- Sử dụng **AsyncStorage** trong React Native
- Sử dụng **Camera** trong React Native


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn