

LẬP TRÌNH ĐA NỀN TẢNG VỚI REACT

BÀI 5: THÀNH PHẦN TRÌNH BÀY PHẦN 1: SỬ DỤNG VIEW VÀ WEBVIEW

- ☐ Sử dụng View trong React Native
- ☐ Sử dụng Webview trong React Native
- ☐ Sử dụng Modal trong React Native
- ☐ Sử dụng ActivityIndicator trong React Native



- ❑ View: Là thành phần phổ biến nhất trong React Native
- ❑ Có thể xem nó như thành phần **div** trong web

- ❑ Các trường hợp sử dụng View:
 - ❖ Khi bạn cần bọc các phần tử bên trong vùng chứa, bạn có thể sử dụng **View** dưới dạng một phần tử vùng chứa (Container Element).

❑ Các trường hợp sử dụng View:

- ❖ Khi bạn muốn tổ hợp nhiều phần tử bên trong phần tử gốc, cả thành phần cha và con đều có thể là View.
- ❖ Nó có thể chứa nhiều thành phần con như bạn muốn.

- ❑ Các trường hợp sử dụng View:
 - ❖ View cũng hỗ trợ các sự kiện cảm ứng tổng hợp, có thể hữu ích cho các mục đích khác nhau.

□ Ví dụ: **src/components/home/Home.js**

```
1 import React, { Component } from 'react'
2 import { View, Text } from 'react-native'
3
4 const Home = () => {
5   return (
6     <View>
7       <View>
8         <Text>This is my text</Text>
9       </View>
10    </View>
11  )
12 }
13 export default Home
```

- ❑ WebView: Nó được sử dụng khi bạn muốn hiển thị trang web trên ứng dụng di động của bạn một cách trực tuyến.

- ❑ Ví dụ sử dụng webview:
- ❑ Tạo và code File
src/components/home/HomeContainer.js

- ❑ Ví dụ sử dụng webview:
- ❑ Bây giờ ta tạo **WebViewExample.js** bên trong thư mục **src/components/home**
- ❑ Code File
src/components/home/WebViewExample.js
- ❑ Hiện nay (tháng 8/2017) react native chưa hỗ trợ webview trên Android, nhưng đã hỗ trợ IOS
- ❑ Có thể dùng webview trên android thông qua framework của Lucas Ferreira

SỬ DỤNG WEBVIEW CỦA LUCAS FERREIRA

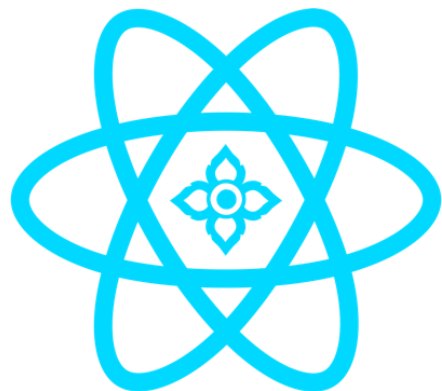
```

var React = require('react-native');
var {
  Platform,
  StyleSheet,
} = React;
var isAndroid = Platform.OS === 'android';

var Example = React.createClass({
  link: 'http://reactplus.com',
  render: function() {
    if (isAndroid) {
      return (
        <WebViewAndroid
          style={styles.webView}
          url={this.link}
        />
      );
    }

    return (
      <WebView
        style={styles.webView}
        url={this.link}
        startInLoadingState={true}
      />
    );
  },
});

```



LẬP TRÌNH ĐA NỀN TẢNG VỚI REACT

BÀI 5: THÀNH PHẦN TRÌNH BÀY **PHẦN 2: SỬ DỤNG MODAL VÀ** **ACTIVITYINDICATOR**

- ❑ Modal: Chúng ta sẽ đặt logic bên trong **ModalExample**.
- ❑ Và có thể cập nhật trạng thái ban đầu bằng cách chạy **toggleModal**.

- ❑ Modal: Sau khi cập nhật trạng thái ban đầu bằng cách chạy **toggleModal**, chúng ta sẽ thiết lập thuộc tính hiển thị cho **Modal** của chúng ta.
- ❑ Yêu cầu này sẽ được cập nhật khi **state** thay đổi.

□ File `src/components/home/Home.js`

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

□ File `src/components/home/ModalExample.is`

```

1  import React, { Component } from 'react';
2
3  import {
4    Modal,
5    Text,
6    TouchableHighlight,
7    View,
8    StyleSheet
9  }
10 from 'react-native'
11
12 class ModalExample extends Component {
13   state = {
14     modalVisible: false,
15   }
16   toggleModal(visible) {
17     this.setState({ modalVisible: visible });
18   }
19   render() {
20     return (
21       <View style = {styles.container}>
22         <Modal animationType = {"slide"} transparent = {false}
23           visible = {this.state.modalVisible}

```


□ File `src/components/home/ModalExample.is`

```

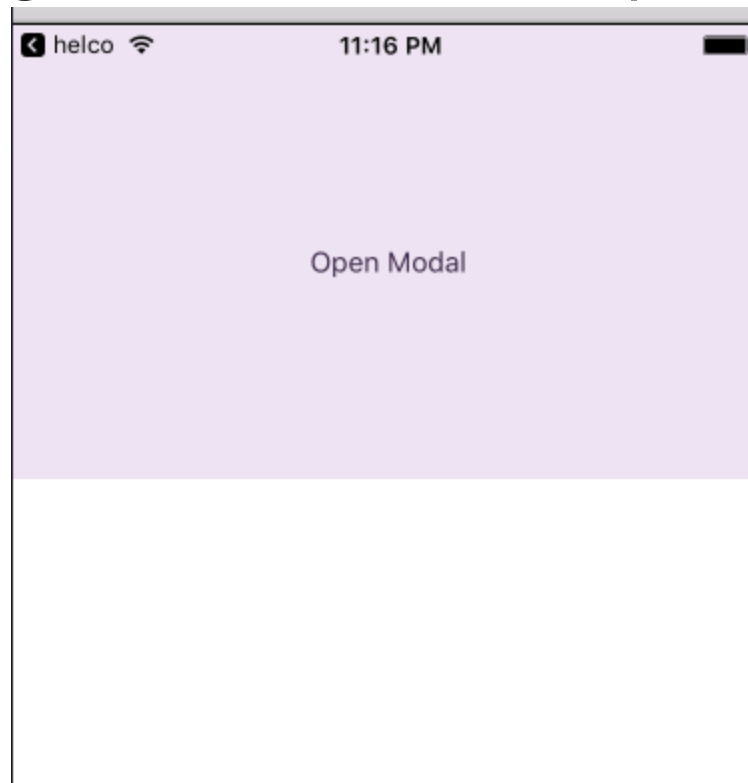
24   onRequestClose = {() => { console.log("Modal has been closed.") } }>
25   <View style = {styles.modal}>
26     <Text style = {styles.text}>Modal is open!</Text>
27
28     <TouchableHighlight onPress = {() => {
29       this.toggleModal(!this.state.modalVisible)}}>
30
31       <Text style = {styles.text}>Close Modal</Text>
32     </TouchableHighlight>
33   </View>
34 </Modal>
35
36   <TouchableHighlight onPress = {() => {this.toggleModal(true)}}>
37     <Text style = {styles.text}>Open Modal</Text>
38   </TouchableHighlight>
39 </View>
40 )
41 }
42 }
43 export default ModalExample
44
45 const styles = StyleSheet.create ({
46   container: {
47     alignItems: 'center',
48     backgroundColor: '#ede3f2',
49     padding: 100
50   },

```

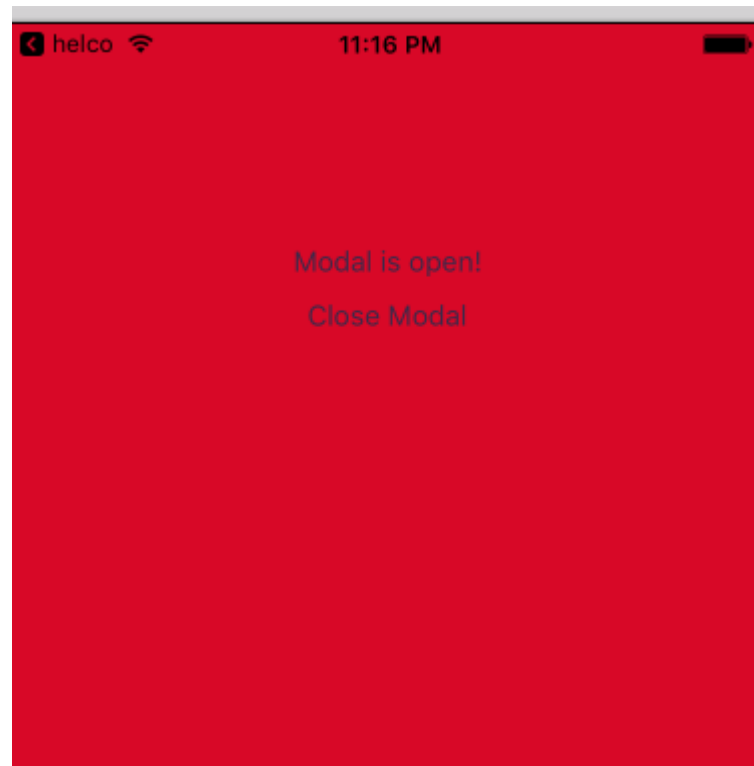
❑ File **src/components/home/ModalExample.js**

```
51 modal: {  
52   flex: 1,  
53   alignItems: 'center',  
54   backgroundColor: '#f7021a',  
55   padding: 100  
56 },  
57 text: {  
58   color: '#3f2949',  
59   marginTop: 10  
60 }  
61 })
```

❑ Chạy chương trình sẽ cho kết quả



□ Khi click vào button, modal sẽ được mở

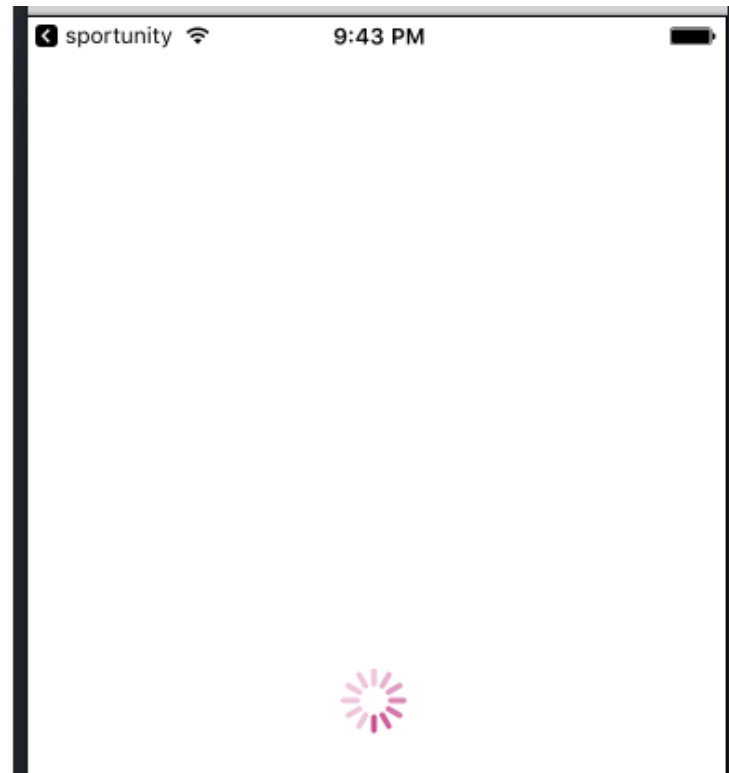


- ❑ Chú ý: Khi xử lý dữ liệu với Modal, dữ liệu sẽ đưa lên server nodejs. Bạn cần lưu ý một số điểm sau:
- ❑ Dữ liệu truyền vào phương thức write có thể xử lý dữ liệu text, hình ảnh,...
- ❑ Khi truyền dữ liệu ta nên sử dụng Stream, kế thừa từ EventEmitter (thiết lập Stream để đọc dữ liệu, cần cấu hình writable bằng true; dùng phương thức createReadStream() cho readable)
- ❑ Để tránh dữ liệu gửi đi nhiều lần, ta cần sử dụng phương thức drain của stream

- ❑ Thuộc tính animation có kiểu Boolean được sử dụng để hiển thị Activity Indicator.
- ❑ Nó sẽ close sau một thời gian đã chọn sau khi các thành phần được gắn kết.
- ❑ Điều này được thực hiện bằng cách sử dụng hàm `closeActivityIndicator()`.

- ❑ Ví dụ: Sử dụng Activity Indicator
- ❑ Ta cần viết code cho các file
src/components/home/Home.js
- ❑ Và
- ❑ **src/components/home/ActivityIndicatorExample.js**

- ❑ Khi chạy ứng dụng, chúng ta sẽ thấy nó load trên màn hình.
- ❑ Nó sẽ biến mất sau vài giây (như cấu hình trong code)



- ❑ Sử dụng View trong React Native
- ❑ Sử dụng Webview trong React Native
- ❑ Sử dụng Modal trong React Native
- ❑ Sử dụng ActivityIndicator trong React Native





Cảm ơn