

## LẬP TRÌNH ĐA NỀN TẢNG VỚI REACT

### BÀI 4: TƯƠNG TÁC VÀ ĐIỀU HƯỚNG

#### PHẦN 1: SỬ DỤNG HTTP VÀ BUTTON

- ☐ Sử dụng Http trong React Native
- ☐ Sử dụng Button trong React Native
- ☐ Sử dụng Animation trong React Native
- ☐ Sử dụng Router trong React Native



- ❑ Http: cách sử dụng tìm nạp (fetch) để xử lý các yêu cầu mạng.
- ❑ Sử dụng Fetch:
- ❑ Chúng ta sẽ sử dụng phương thức vòng đời **componentDidMount** để tải dữ liệu từ máy chủ ngay khi thành phần được gắn kết.

- ❑ Http: cách sử dụng tìm nạp (fetch) để xử lý các yêu cầu mạng.
- ❑ Sử dụng Fetch:
- ❑ Chức năng này sẽ gửi yêu cầu GET đến máy chủ, trả lại dữ liệu JSON, đăng xuất đầu ra để điều khiển và cập nhật trạng thái.

## □ File `src/components/home/Home.js`

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

## □ File src/components/home/HttpExample.is

```
1 import React, { Component } from 'react'
2 import { View, Text } from 'react-native'
3
4 class HttpExample extends Component {
5   state = {
6     data: ''
7   }
8   componentDidMount = () => {
9     fetch('https://jsonplaceholder.typicode.com/posts/1', {
10       method: 'GET'
11     })
12     .then((response) => response.json())
13     .then((responseJson) => {
14       console.log(responseJson);
15
16       this.setState({
17         data: responseJson
18       })
19     })
20     .catch((error) => {
21       console.error(error);
22     });
23   }
```

## □ File `src/components/home/HttpExample.js`

```
24   render() {  
25     return (  
26       <View>  
27         <Text>  
28           {this.state.data.body}  
29         </Text>  
30       </View>  
31     )  
32   }  
33 }  
34 export default HttpExample
```

- ❑ Button: Chúng ta thấy các thành phần có thể chạm vào React Native là button.
- ❑ Button có thể gọi là "touchable" bởi vì chúng cung cấp hình ảnh động được xây dựng
- ❑ Có thể sử dụng tính năng onPress để xử lý sự kiện chạm.



- ❑ Facebook cung cấp thành phần Button, có thể được sử dụng như một generic button
- ❑ Nếu thành phần Button mặc định không phù hợp với nhu cầu của bạn, bạn có thể sử dụng một trong các thành phần sau.
- ❑ **Touchable Opacity**
- ❑ **Touchable Highlight**
- ❑ **Touchable Native Feedback**
- ❑ **Touchable Without Feedback**

## □ **Touchable Opacity:**

❖ Phần tử này sẽ thay đổi độ mờ của một phần tử khi chạm vào.

❖ Code file

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

## **Touchable Highlight**

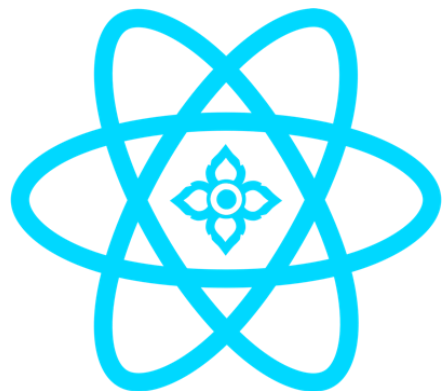
- ❖ Khi người dùng nhấn vào phần tử, nó sẽ trở nên tối hơn và màu nền sẽ hiển thị.

## **Touchable Native Feedback**

❖ Điều này sẽ mô phỏng hoạt hình khi thành phần được nhấn.

## ❑ **Touchable Without Feedback**

- ❖ Điều này nên được sử dụng khi bạn muốn xử lý các sự kiện liên lạc mà không có hiệu ứng hoạt hình
- ❖ Thông thường, thành phần này không được sử dụng nhiều.



## **LẬP TRÌNH ĐA NỀN TẢNG VỚI REACT**

### **BÀI 4: TƯƠNG TÁC VÀ ĐIỀU HƯỚNG PHẦN 2: SỬ DỤNG ANIMATION VÀ ROUTER**

- ❑ Animations là một thành phần rất quan trọng để tạo nên sự tuyệt vời trong trải nghiệm người dùng.
- ❑ Ví như các vật thể cố định phải vượt qua quán tính đứng yên để bắt đầu di chuyển.
- ❑ Các đối tượng chuyển động có đà và hiếm khi dừng ngay lập tức. Animations cho phép bạn chuyển tải các chuyển động tương tự các vật thể trong giao diện của ứng dụng.
- ❑ React Native cung cấp hai hệ thống animation: Animation để kiểm soát chi tiết và tương tác với các giá trị cụ thể và LayoutAnimation để di chuyển giao diện tổng thể.

- ❑ Animation: Hỗ trợ sử dụng **LayoutAnimation** trong React Native
- ❑ Ta sẽ import thành phần Animation và render nó trên màn hình Home
- ❑ Code file **src/components/home/Animations.js**



## □ Code file

### src/components/home/Animations.js

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

## ❑ Animations Component

- ❖ Chúng ta sẽ thiết lập myStyle làm props của state.
- ❖ Thuộc tính này được sử dụng để tạo kiểu dáng một phần tử bên trong PresentationalAnimationComponent.

## ❑ Animations Component

- ❖ Chúng ta cũng sẽ tạo ra hai chức năng - `expandElement` và `collapseElement`.
- ❖ Các chức năng này sẽ cập nhật các giá trị từ **state**.

## ❑ Animations Component

- ❖ Đầu tiên sẽ sử dụng hình ảnh Animation sẵn có ở phần tài nguyên, tiếp theo hình ảnh thứ hai sẽ là hình ảnh tĩnh.
- ❖ Chúng ta sẽ gọi các hàm **expandElement()** và **collapseElement()**.

## ❑ Animations Component

- ❖ Đầu tiên sẽ sử dụng hình ảnh Animation sẵn có ở phần tài nguyên, tiếp theo hình ảnh thứ hai sẽ là hình ảnh tĩnh.
- ❖ Chúng ta sẽ gọi các hàm **expandElement()** và **collapseElement()**.

## ❑ File `src/components/home/Home`

```
1 import React, { Component } from 'react'
2 import { View, LayoutAnimation, TouchableOpacity, Text, StyleSheet }
3 from 'react-native'
4
5 class Animations extends Component {
6   state = {
7     myStyle: {
8       height: 100,
9       backgroundColor: 'red'
10    }
11  }
12  expandElement = () => {
13    LayoutAnimation.configureNext(LayoutAnimation.Presets.spring);
14    this.setState({
15      myStyle: {
16        height: 400,
17        backgroundColor: 'red'
18      }
19    })
20  }
```

# CÁC KHÁI NIỆM: ANIMATION

```

21 collapseElement = () => {
22     LayoutAnimation.configureNext(LayoutAnimation.Presets.linear);
23
24     this.setState({
25         myStyle: {
26             height: 100,
27             backgroundColor: 'red'
28         }
29     })
30 }
31 render() {
32     return (
33         <View>
34             <View>
35                 <View style = {this.state.myStyle}>
36                 </View>
37
38                 <TouchableOpacity>
39                     <Text style = {styles.button} onPress = {this.expandElement}>
40                         Expand
41                     </Text>
42                 </TouchableOpacity>
43
44                 <TouchableOpacity>
45                     <Text style = {styles.button} onPress = {this.collapseElement}>
46                         Collapse
47                     </Text>
48                 </TouchableOpacity>

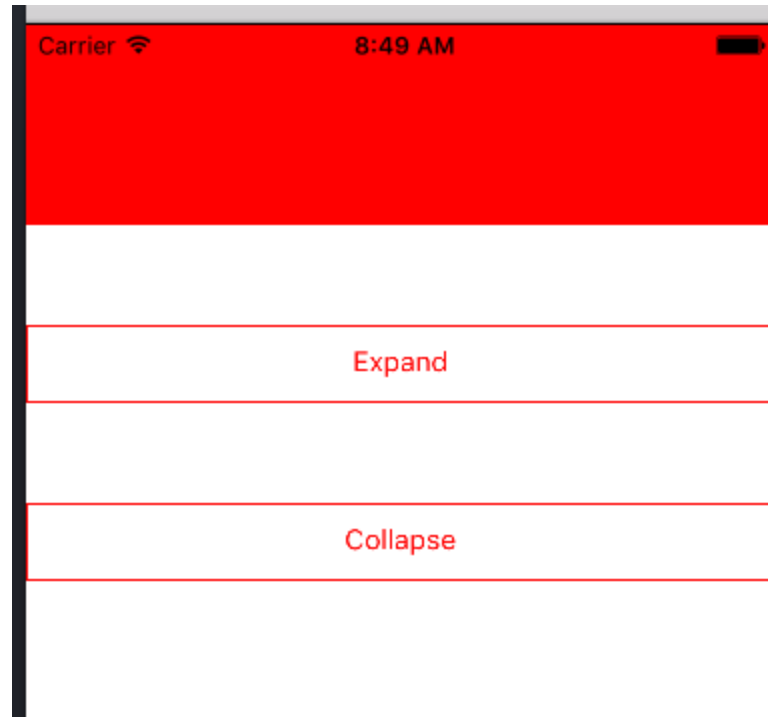
```

## ❑ File `src/components/home/Home`

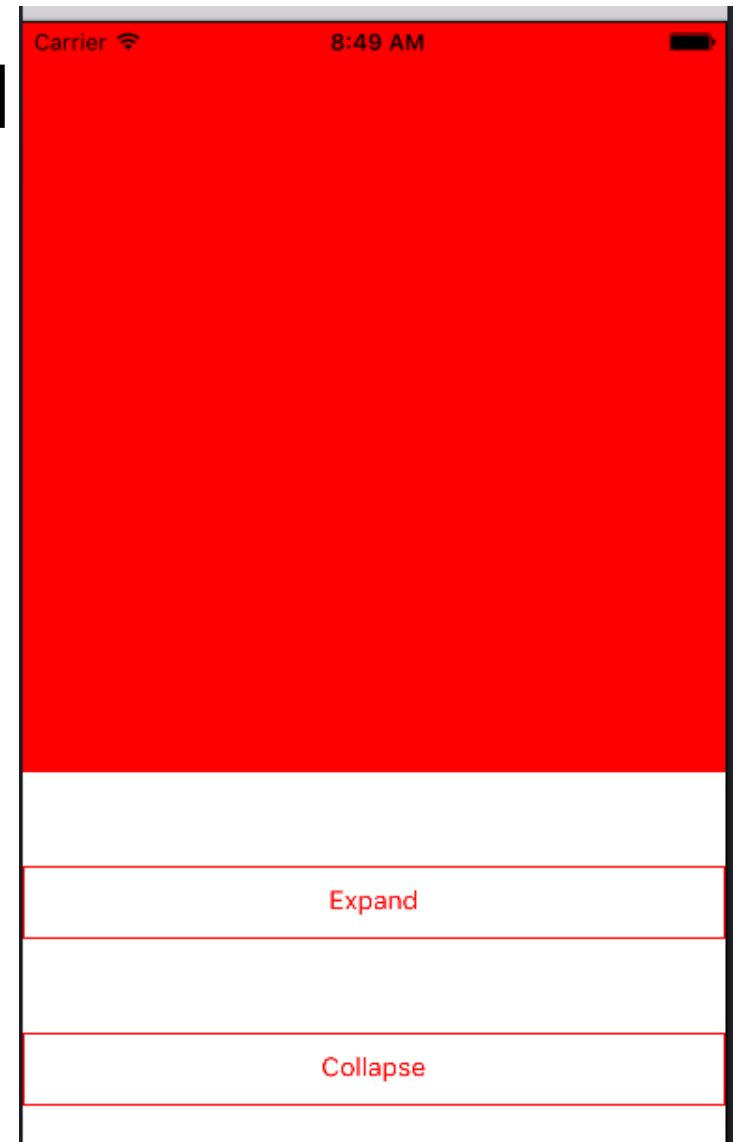
```
49         </View>
50     )
51 }
52 }
53 export default Animations
54
55 const styles = StyleSheet.create({
56     button: {
57         borderWidth: 1,
58         borderColor: 'red',
59         color: 'red',
60         textAlign: 'center',
61         marginTop: 50,
62         padding: 10
63     }
64 })
65 export default Home
```



## ❑ Chạy chương trình cho kết quả



□ Khi click vào nút expand



- ❑ Router: Hỗ trợ navigation trong React Native
- ❑ Các bước:
  - ❖ **Install Router**
  - ❖ **Cấu hình index.android.js.**
  - ❖ **Add Router**
  - ❖ **Create components**

## ❑ Bước 1: **Install Router**

❖ Dùng câu lệnh:

➤ `npm i react-native-router-flux –save`

## ❑ Bước 2: cài đặt **index.android.js**.

## ❑ Bước 3: **Add Router**

❖ Code File

**src/components/routes/Routes.js**

## ❑ Bước 4: **Create components**

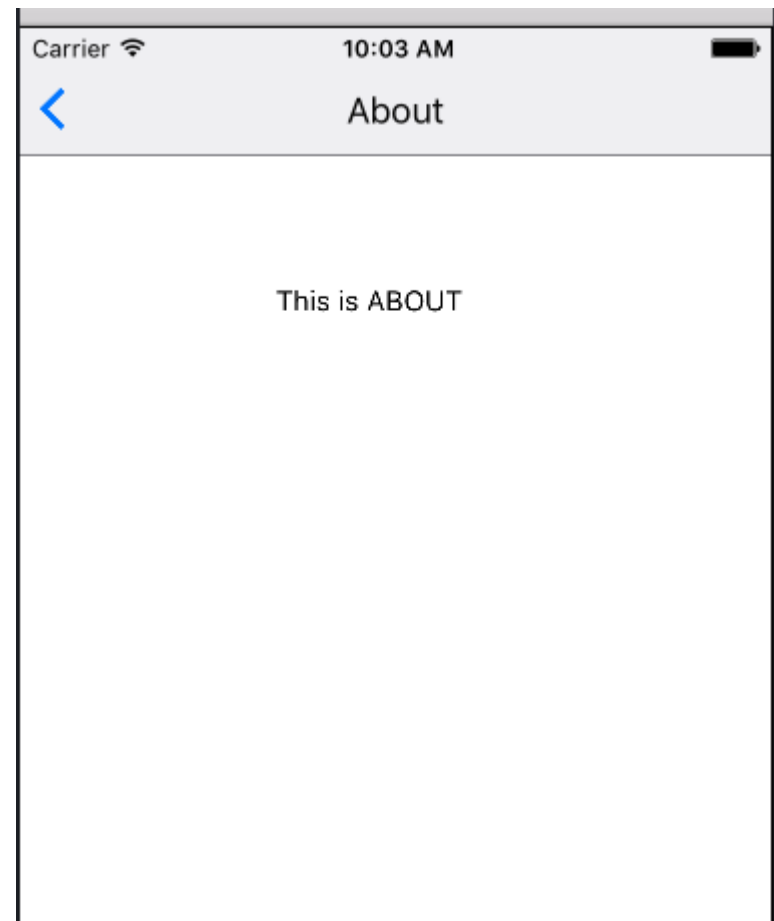
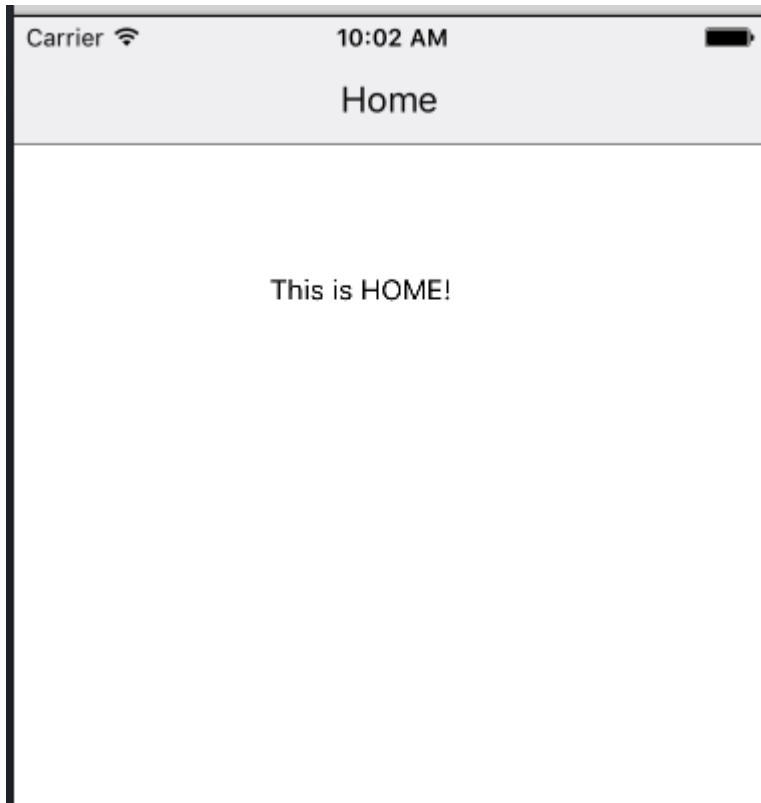
❖ Code File

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

❖ Code File

**src/components/home/About.js**

## □ Hình ảnh về Router



- ☐ Sử dụng Http trong React Native
- ☐ Sử dụng Button trong React Native
- ☐ Sử dụng Animation trong React Native
- ☐ Sử dụng Router trong React Native





**Cảm ơn**