

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

### **BÀI 6: INPUT VÀ SUBMIT DỮ LIỆU**

#### **PHẦN 1: SỬ DỤNG PICKER VÀ STATUSBAR**

- ☐ Sử dụng Picker trong React Native
- ☐ Sử dụng Statusbar trong React Native



- ❑ Picker: Sử dụng để lựa chọn giá trị hiển thị trong React Native
- ❑ Ví dụ: Ta sẽ tạo ra Picker đơn giản với hai tùy chọn có sẵn
- ❑ Sử dụng Switch trong React Native
- ❑ Sử dụng Text trong React Native

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

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

## ❑ Logic hoạt động:

- ❖ `this.state.user` được sử dụng để kiểm soát picker.
- ❖ Chức năng `updateUser` sẽ được kích hoạt khi người dùng chọn một giá trị.
- ❖ Xem file  
**`src/components/home/PickerExample.js`**

❑ file `src/components/home/PickerExample.is`

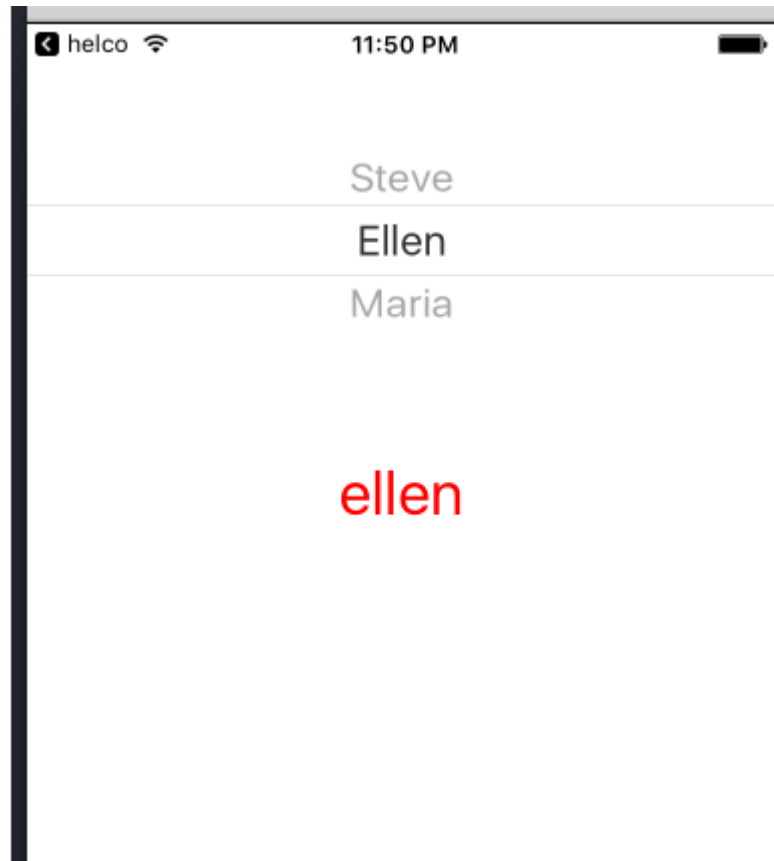
```

1  import React, { Component } from 'react';
2  import { View, Text, Picker, StyleSheet } from 'react-native'
3
4  class PickerExample extends Component {
5    state = {user: ''}
6    updateUser = (user) => {
7      this.setState({ user: user })
8    }
9    render() {
10     return (
11       <View>
12         <Picker selectedValue = {this.state.user}
13         |onValueChange = {this.updateUser}>
14           <Picker.Item label = "Steve" value = "steve" />
15           <Picker.Item label = "Ellen" value = "ellen" />
16           <Picker.Item label = "Maria" value = "maria" />
17         </Picker>
18         <Text style = {styles.text}>{this.state.user}</Text>
19       </View>
20     )
21   }
22 }
  
```

## ❑ file `src/components/home/PickerExample.js`

```
23 export default PickerExample
24
25 const styles = StyleSheet.create({
26   text: {
27     fontSize: 30,
28     alignSelf: 'center',
29     color: 'red'
30   }
31 })
```

□ Khi chạy chương trình sẽ cho kết quả như sau:

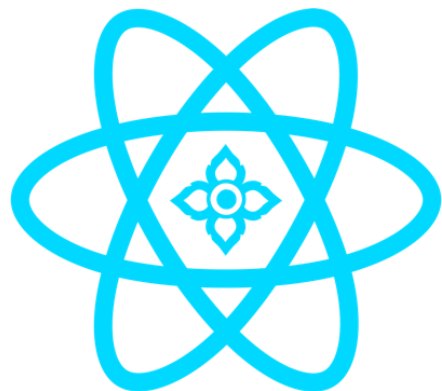




- ❑ Khi cập nhật dữ liệu Picker lên server nodejs cần chú ý:
- ❑ Sau khi submit, stream data bị đóng và ta cần gọi sự kiện close để giải phóng kết nối.
- ❑ Nếu người dùng lựa chọn giá trị nhưng không submit, ta cần tạm dừng stream bằng phương thức `process.stdin.pause()`
- ❑ Khi người dùng đã chọn xong, không cần thay đổi gì thì cần gọi sự kiện finish của stream
- ❑ Sau khi submit toàn bộ dữ liệu xong, cần gọi sự kiện `end()`

- ☐ Statusbar dễ sử dụng và bạn chỉ cần thiết lập thuộc tính để thay đổi nó.
- ☐ Thuộc tính ẩn có thể được sử dụng để ẩn thanh trạng thái.
- ☐ Mặc định nó được đặt thành false.

- ❑ BarStyle có thể có ba giá trị - dark-content, light-content và default.
- ❑ Thành phần này có một số tính chất khác có thể được sử dụng.
- ❑ StatusBar là một component trong react native



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

### **BÀI 6: INPUT VÀ SUBMIT DỮ LIỆU**

#### **PHẦN 2: SỬ DỤNG SWITCH VÀ TEXT**

- ❑ Giống như radio button, Switch trong React Native dùng để chọn một trong hai giá trị



- ❑ Ví dụ: Tạo thành phần Switch
- ❑ Logic hoạt động:
  - ❖ Chúng ta sẽ chuyển giá trị từ state và các chức năng để chuyển các items cần chuyển đổi thành thành phần của SwitchExample.
  - ❖ Chức năng Toggle sẽ được sử dụng để cập nhật trạng thái.

- ❑ Trước tiên ta tạo file  
**src/components/home/SwitchExample.js**
- ❑ Tiếp theo viết code cho file  
**src/component/home/Home.js**

# THÀNH PHẦN VÀ API: SWITCH

 file src/component/home/Home.js

```

home.js
1  import React, { Component } from 'react'
2  import {
3    View
4  }
5  from 'react-native'
6  import SwitchExample from './SwitchExample'
7
8  export default class HomeContainer extends Component {
9    constructor() {
10      super();
11      this.state = {
12        switch1Value: false,
13        switch2Value: false,
14      }
15    }
16    toggleSwitch1 = (value) => {
17      this.setState({switch1Value: value})
18      console.log('Switch 1 is: ' + value)
19    }
20    toggleSwitch2 = (value) => {
21      this.setState({switch2Value: value})
22      console.log('Switch 2 is: ' + value)
23    }
  
```



## ❑ file `src/component/home/Home.js`

```
24  render() {  
25      return (  
26          <View>  
27              <SwitchExample  
28                  toggleSwitch1 = {this.toggleSwitch1}  
29                  toggleSwitch2 = {this.toggleSwitch2}  
30                  switch1Value = {this.state.switch1Value}  
31                  switch2Value = {this.state.switch2Value}/>  
32              </View>  
33          );  
34      }  
35  }
```

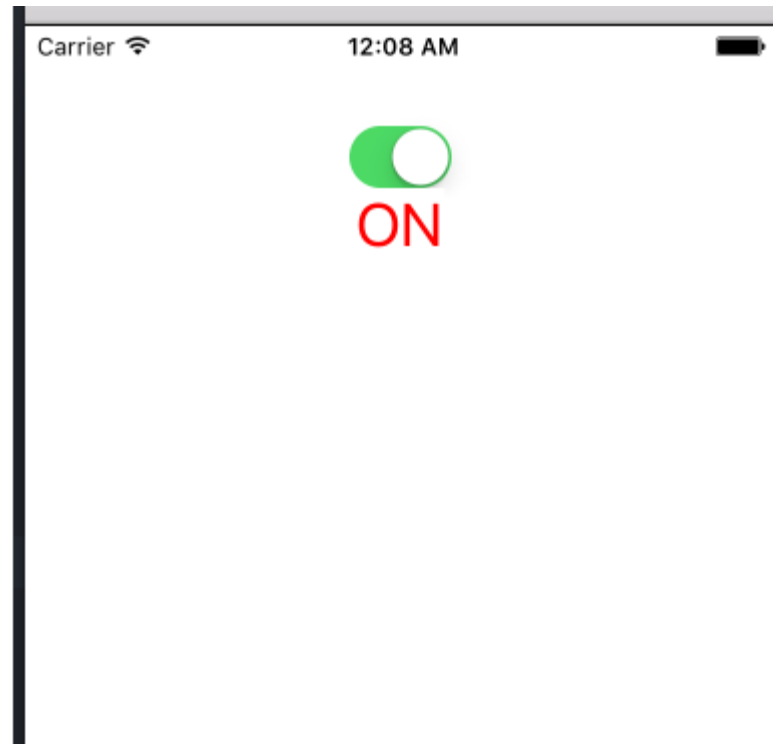
- ❑ Thành phần Switch có hai props.
- ❑ Các prop `onValueChange` sẽ kích hoạt các chức năng Switch sau khi người dùng nhấn nút switch.
- ❑ Giá trị prop được ràng buộc với trạng thái của thành phần `HomeContainer`.

□ Cụ thể như đoạn code sau:

```

1  import React, { Component } from 'react'
2  import {
3    View,
4    Switch,
5    StyleSheet
6  }
7  from 'react-native'
8
9  export default SwitchExample = (props) => {
10    return (
11      <View style = {styles.container}>
12        <Switch
13          onValueChange = {props.toggleSwitch1}
14          value = {props.switch1Value}/>
15        <Switch
16          onValueChange = {props.toggleSwitch2}
17          value = {props.switch2Value}/>
18      </View>
19    )
20  }
21  const styles = StyleSheet.create ({
22    container: {
23      flex: 1,
24      alignItems: 'center',
25      marginTop: 100
26    }
27  })
  
```

□ Khi chạy chương trình sẽ cho ra kết quả



- ❑ Chú ý: Khi lựa chọn Switch, giá trị chọn sẽ được React cập nhật lên server nodejs, ta cần nhắc lại một số khái niệm
- ❑ Nếu dữ liệu cần cập nhật được đọc từ file, ta sử dụng phương thức `path()` để cập nhật đường dẫn file
- ❑ Vì một lý do nào đó, file bị xóa hoặc không tồn tại, ta cần tạo file mới, sử dụng cờ `w+`; hoặc bạn đưa ra ngoại lệ bằng cờ `r+` khi không muốn chương trình tạo file mới
- ❑ Nếu xảy ra lỗi ta cần sử dụng Stream `stderr`

- ❑ Text: Thành phần này có thể được lồng nhau và nó có thể kế thừa các thuộc tính từ cha sang con.
- ❑ Điều này có thể hữu ích và được sử dụng bằng nhiều cách.
- ❑ Ví dụ: viết hoa chữ cái đầu tiên, tạo kiểu chữ hoặc các phần của văn bản, v.v.

- ❑ Chúng ta cần sử dụng pattern kế thừa cho text.
- ❑ `styles.text` sẽ được áp dụng cho tất cả các thành phần Text.
- ❑ Bạn chú ý cách thiết lập các thuộc tính kiểu dáng khác cho một số phần của văn bản.
- ❑ Điều quan trọng là phải biết rằng tất cả các thành phần con có style giống thành phần cha.

- ❑ Picker: Sử dụng để lựa chọn giá trị hiển thị trong React Native
- ❑ Ví dụ: Ta sẽ tạo ra Picker đơn giản với hai tùy chọn có sẵn
- ❑ Sử dụng Switch trong React Native
- ❑ Sử dụng Text trong React Native







**Cảm ơn**