

วิธีการใช้งาน Rating, Checkbox, Radio Buttons, และ Switch ใน React Native

Rating

ติดตั้ง:

npm install react-native-ratings

```
import React from 'react';
import { View, StyleSheet } from 'react-native';
import { AirbnbRating } from 'react-native-ratings';

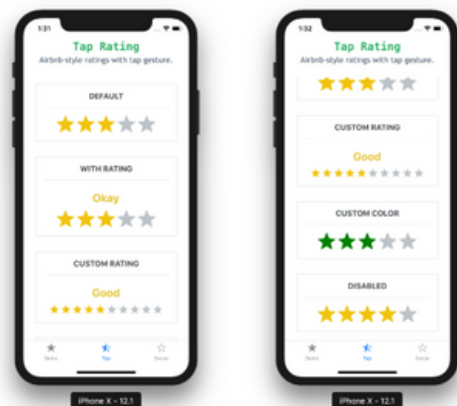
const RatingExample = () => {
  const handleRating = (rating) => {
    console.log("คะแนนที่เลือก:", rating);
  };

  return (
    <View style={styles.container}>
      <AirbnbRating
        count={5}
        reviews={['แย่มาก', 'พอใช้', 'ดี', 'ดีมาก', 'ยอดเยี่ยม']}
        defaultRating={3}
        size={30}
        onFinishRating={handleRating}
      />
    </View>
  );
};

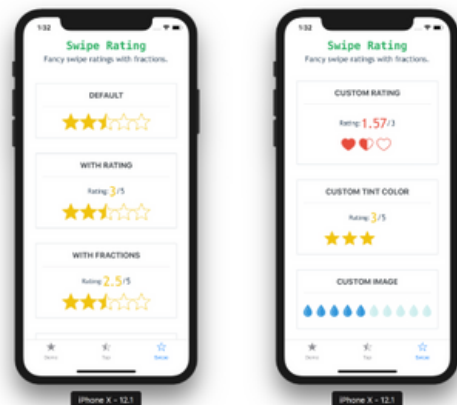
const styles = StyleSheet.create({
  container: {
    marginTop: 50
  }
});

export default RatingExample;
```

Tap Rating



Swipe Rating



พรีอพ

รายละเอียด

count

จำนวนดาว (ปกติ 5)

reviews

คำอธิบายแต่ละระดับดาว

defaultRating

ค่าเริ่มต้นของดาวที่เลือก

size

ขนาดของดาว

onFinishRating

ฟังก์ชันที่ทำงานเมื่อผู้ใช้ให้คะแนนเสร็จ

Checkbox

ติดตั้ง:

npm install react-native-elements react-native-vector-icons

```
import React, { useState } from 'react';
import { View } from 'react-native';
import { CheckBox } from 'react-native-elements';

const CheckboxExample = () => {
  const [checked, setChecked] = useState(false);

  return (
    <View style={{ marginTop: 50 }}>
      <CheckBox
        title="ฉันยอมรับเงื่อนไขการใช้งาน"
        checked={checked}
        onPress={() => setChecked(!checked)}
      />
    </View>
  );
};

export default CheckboxExample;
```

☐ ฉันยอมรับเงื่อนไขการใช้งาน

☒ ฉันยอมรับเงื่อนไขการใช้งาน

(หาก checked = true จะมีเครื่องหมายถูก)

title ข้อความแสดงข้างกล่องติ๊ก
checked สถานะการติ๊ก (true/false)
onPress ฟังก์ชันเมื่อผู้ใช้แตะ Checkbox

Radio Buttons

ติดตั้ง:

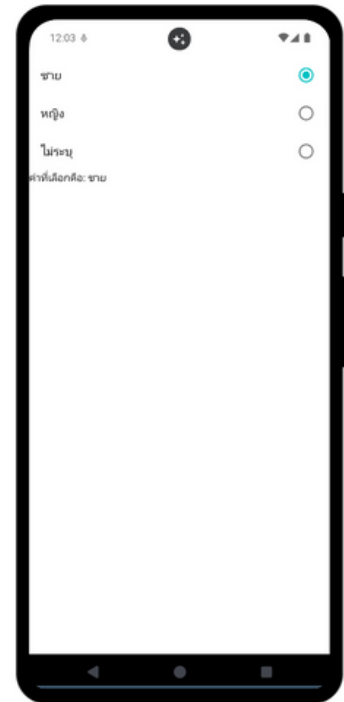
npm install react-native-paper

```
import React, { useState } from 'react';
import { View } from 'react-native';
import { RadioButton, Text } from 'react-native-paper';

const RadioExample = () => {
  const [value, setValue] = useState('ชาย');

  return (
    <View style={{ marginTop: 50 }}>
      <RadioButton.Group onChange={newValue => setValue(newValue)} value={value}>
        <RadioButton.Item label="ชาย" value="ชาย" />
        <RadioButton.Item label="หญิง" value="หญิง" />
        <RadioButton.Item label="ไม่ระบุ" value="ไม่ระบุ" />
      </RadioButton.Group>
      <Text>ค่าที่เลือกคือ: {value}</Text>
    </View>
  );
};

export default RadioExample;
```



RadioButton.Group กลุ่มปุ่มที่เลือกได้เพียงหนึ่งค่า
value ค่าที่ถูกเลือก
onChange ฟังก์ชันที่ทำงานเมื่อมีการเปลี่ยนค่า
RadioButton.Item ตัวเลือกแต่ละตัวพร้อมข้อความแสดง

ใช้ในกรณีใด:

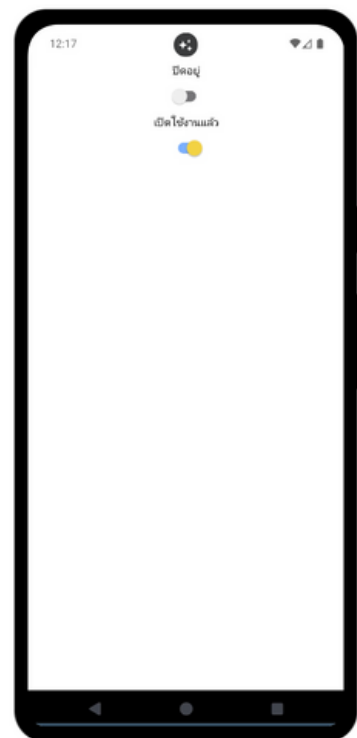
- แบบฟอร์มเลือกเพศ
- แบบฟอร์มเลือกช่วงอายุ
- เลือกช่องทางการติดต่อ เช่น โทรศัพท์ / อีเมล / แชท

Switch

คอมโพเนนต์ Switch มีอยู่ใน React Native อยู่แล้ว

ใช้งานง่าย เหมาะกับกรณีเปิด/ปิดฟีเจอร์ เช่น การแจ้งเตือน, โหมดมืด, หรือสถานะต่าง ๆ

```
1 import React, { useState } from 'react';
2 import { View, Text, Switch, StyleSheet } from 'react-native';
3
4 const SwitchExample = () => {
5   const [isEnabled, setIsEnabled] = useState(false);
6   const [isEnabledT, setIsEnabledT] = useState(false);
7
8   const toggleSwitch = () => setIsEnabled(previousState => !previousState);
9
10  return (
11    <View style={styles.container}>
12      <Text>{isEnabled ? 'เปิดใช้งานแล้ว' : 'ปิดอยู่'}</Text>
13      <Switch
14        value={isEnabled}
15        onValueChange={toggleSwitch}
16        trackColor={{ false: '#767577', true: '#81b0ff' }}
17        thumbColor={isEnabled ? '#f5dd4b' : '#f4f3f4'}
18      />
19      <Text>{isEnabledT ? 'เปิดใช้งานแล้ว' : 'ปิดอยู่'}</Text>
20      <Switch
21        value={isEnabledT}
22        onValueChange={setIsEnabledT}
23        trackColor={{ false: '#767577', true: '#81b0ff' }}
24        thumbColor={isEnabledT ? '#f5dd4b' : '#f4f3f4'}
25      />
26    </View>
27  );
28 };
29
30 const styles = StyleSheet.create({
31   container: {
32     marginTop: 50,
33     alignItems: 'center'
34   }
35 });
36
37 export default SwitchExample;
```



value ค่าปัจจุบันของ Switch (เปิด/ปิด)
onValueChange ฟังก์ชันเมื่อสถานะเปลี่ยน
trackColor สีพื้นหลังของแถบ (เปิด/ปิด)
thumbColor สีของปุ่มเลื่อน

ใช้ในกรณี:

เปิด/ปิดการแจ้งเตือน

เปิด/ปิดโหมดมืด

เปิด/ปิดการอนุญาต (ตำแหน่ง, กล้อง ฯลฯ)