**Statefulwidget trong Flutter**

Trong bài trước mình đã giới thiệu có hai phương thức quản lý trạng thái trong **Flutter** đó là quản lý trạng thái ngắn hạn và quản lý trạng thái ứng dụng. Trong bài này chúng ta sẽ tìm hiểu quản lý trạng thái ngắn hạn với StatefulwidgetTrong bài trước mình đã giới thiệu có hai phương thức quản lý trạng thái trong Flutter đó là quản lý trạng thái ngắn hạn và quản lý trạng thái ứng dụng. Trong bài này chúng ta sẽ tìm hiểu quản lý trạng thái ngắn hạn với **Statefulwidget**

Vì ứng dụng Flutter được tạo nên từ các widget, do đo sviệc quản lý trạng thái cũng được thực hiện bởi widget. Mấu chốt của việc quản lý trạng thái là *Statefulwidget*. Widget được kết thừa từ Statefulwidget để duy trì trạng thái và quản lý các trạng thái con của nó.

Statefulwidget cung cấp tuỳ chọn cho widget để tạo ra các trạng thái, việc khởi tạo trang thái ban đầu của widget được thực hiện qua hàm *createState* và hàm *setState* dùng để thay đổi trạng thái khi cần. Sự thay đổi trạng thái được thực hiện qua các gesture (cử chỉ). Ví dụ, sự đánh giá (rating) của một sản phẩm có thể được thay đổi khi người dùng chạm vào sao của rating widget.

Bây giờ chúng ta sẽ tạo một widget là RatingBox để làm quen với quản lý trạng thái. Mục đích của widget là hiển thị số lượng đánh giá hiện tại của sản phẩm.

Bước đầu tiên chúng ta tạo một widget là RatingBox kế thừa từ StatefulWidget.

class RatingBox extends StatefulWidget { }

Tạo một state cho RatingBox là  \_RatingBoxState kế thừa từ  State<T>

class \_RatingBoxState extends State<RatingBox> { }

Ghi đè phương thức createState của StatefulWidget để tạo trang thái cho widget bằng \_RatingBoxState

class RatingBox extends StatefulWidget {

@override

\_RatingBoxState createState() => \_RatingBoxState();

}

Tạo giao diện người dùng cho RatingBox widget bằng phương thức build của \_RatingBoxState. Thông thường chúng ta sẽ tạo giao diện bằng phương thức build của RatingBox widget. Tuy nhiên khi cần quản lý trạng thái chúng ta cần tạo giao diện ở trong \_RatingBoxState widget. Điều này đảm bảo cho việc hiển thị lại giao diện người dùng khi trạng thái của widget thay đổi.

Widget build(BuildContext context) {

double \_size = 20;

print(\_rating);

return Row(

mainAxisAlignment: MainAxisAlignment.end,

crossAxisAlignment: CrossAxisAlignment.end,

mainAxisSize: MainAxisSize.max,

children: <Widget>[

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 1 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

iconSize: \_size,

),

), Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 2 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

iconSize: \_size,

),

), Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 3 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

iconSize: \_size,

),

),

],

);

}

Ở đây, chúng ta sử dụng 3 ngôi sao, được tạo nên từ IconButton widget và sắp xếp sử dụng Row widget trên cùng 1 hàng ngang. Ý tưởng ở đây là hiển thị rating thông qua dãy các ngôi sao. Ví dụ nếu rating của sản phẩm là 2 sao thì hai ngôi sao đầu sẽ có màu đỏ và ngôi sao cuối cùng sẽ có màu trắng.

Thêm phương thức thay đổi trạng thái cho \_RatingBoxState

void \_setRatingAsOne() {

setState( () {

\_rating = 1;

});

}

void \_setRatingAsTwo() {

setState( () {

\_rating = 2;

});

}

void \_setRatingAsThree() {

setState( () {

\_rating = 3;

});

}

Thay đổi trạng thái của rating theo cử chỉ của người dùng (chạm vào ngôi sao)

Widget build(BuildContext context) {

double \_size = 20;

print(\_rating);

return Row(

mainAxisAlignment: MainAxisAlignment.end,

crossAxisAlignment: CrossAxisAlignment.end,

mainAxisSize: MainAxisSize.max,

children: <Widget>[

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 1 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsOne,

iconSize: \_size,

),

),

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 2 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsTwo,

iconSize: \_size,

),

),

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 3 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsThree,

iconSize: \_size,

),

),

],

);

}

Ở đây sự kiện onPressed sẽ gọi hàm tương ứng để thay đổi trạng thái của widget và hiển thị ra giao diện. Khi trạng thái bị thay đổi, hàm buid sẽ được gọi lại và giao diện sẽ được hiển thị lại.

Toàn bộ code của RatingBox widget như sau:

class RatingBox extends StatefulWidget {

@override

\_RatingBoxState createState() => \_RatingBoxState();

}

class \_RatingBoxState extends State<RatingBox> {

int \_rating = 0;

void \_setRatingAsOne() {

setState( () {

\_rating = 1;

});

}

void \_setRatingAsTwo() {

setState( () {

\_rating = 2;

});

}

void \_setRatingAsThree() {

setState( () {

\_rating = 3;

});

}

Widget build(BuildContext context) {

double \_size = 20;

print(\_rating);

return Row(

mainAxisAlignment: MainAxisAlignment.end,

crossAxisAlignment: CrossAxisAlignment.end,

mainAxisSize: MainAxisSize.max,

children: <Widget>[

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 1 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsOne,

iconSize: \_size,

),

),

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 2 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsTwo,

iconSize: \_size,

),

),

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 3 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsThree,

iconSize: \_size,

),

),

],

);

}

}

Bây giờ ta sẽ chỉnh sửa lại ứng dụng product\_layout\_app ở bài trước bằng cách bổ sung RatingBox widget và thêm vào ProductBox như sau:

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo', theme: ThemeData(

primarySwatch: Colors.blue,),

home: MyHomePage(title: 'Product layout demo home page'),

);

}

}

class MyHomePage extends StatelessWidget {

MyHomePage({Key key, this.title}) : super(key: key);

final String title;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title:Text("Product Listing")),

body: ListView(

shrinkWrap: true, padding: const EdgeInsets.fromLTRB(2.0, 10.0, 2.0, 10.0),

children: <Widget> [

ProductBox(

name: "iPhone",

description: "iPhone is the stylist phone ever",

price: 1000,

image: "iphone.jpg"

),

ProductBox(

name: "Pixel",

description: "Pixel is the most featureful phone ever",

price: 800,

image: "pixel.jpg"

),

ProductBox(

name: "Laptop",

description: "Laptop is most productive development tool",

price: 2000,

image: "laptop.jpg"

),

ProductBox(

name: "Tablet",

description: "Tablet is the most useful device ever for meeting",

price: 1500,

image: "tablet.jpg"

),

ProductBox(

name: "Pendrive",

description: "Pendrive is useful storage medium",

price: 100,

image: "pendrive.jpg"

),

ProductBox(

name: "Floppy Drive",

description: "Floppy drive is useful rescue storage medium",

price: 20,

image: "floppydisk.jpg"

),

],

)

);

}

}

class RatingBox extends StatefulWidget {

@override

\_RatingBoxState createState() =>

\_RatingBoxState();

}

class \_RatingBoxState extends State<RatingBox> {

int \_rating = 0;

void \_setRatingAsOne() {

setState( () {

\_rating = 1;

});

}

void \_setRatingAsTwo() {

setState( () {

\_rating = 2;

});

}

void \_setRatingAsThree() {

setState( () {

\_rating = 3;

});

}

Widget build(BuildContext context) {

double \_size = 20;

print(\_rating);

return Row(

mainAxisAlignment: MainAxisAlignment.end,

crossAxisAlignment: CrossAxisAlignment.end,

mainAxisSize: MainAxisSize.max,

children: <Widget>[

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 1 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsOne,

iconSize: \_size,

),

),

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 2 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsTwo,

iconSize: \_size,

),

),

Container(

padding: EdgeInsets.all(0),

child: IconButton(

icon: (\_rating >= 3 ? Icon(Icons.star, size: \_size,) :

Icon(Icons.star\_border, size: \_size,)),

color: Colors.red[500],

onPressed: \_setRatingAsThree,

iconSize: \_size,

),

),

],

);

}

}

class ProductBox extends StatelessWidget {

ProductBox({Key key, this.name, this.description, this.price, this.image}) :

super(key: key);

final String name;

final String description;

final int price;

final String image;

Widget build(BuildContext context) {

return Container(

padding: EdgeInsets.all(2),

height: 140,

child: Card(

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: <Widget>[

Image.asset("assets/appimages/" + image),

Expanded(

child: Container(

padding: EdgeInsets.all(5),

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: <Widget>[

Text(this.name, style: TextStyle(fontWeight: FontWeight.bold)),

Text(this.description),

Text("Price: " + this.price.toString()),

RatingBox(),

],

)

)

)

]

)

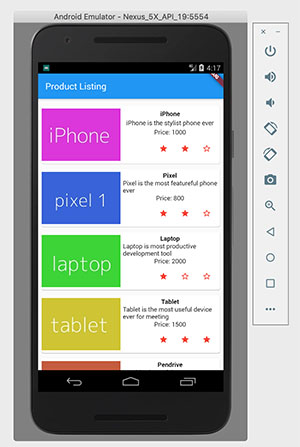
)

);

}

}

Các bạn chạy thử ứng dụng, và chạm vào các ngôi sao để xem trạng thái của RatingBox thay đổi



Như vậy trong bài này mình đã hướng dẫn các bạn làm quen với việc quản lý trạng thái của widget thông qua StatefulWidget. Chúc các bạn học tốt