# 3.6 单选开关和复选框

Material 组件库中提供了Material风格的单选开关Switch和复选框Checkbox，它们都是继承自StatelessWidget，所以它们本身不会保存当前选择状态，因此它们的选中状态都是由父组件来管理的。当Switch或Checkbox被点击时，会触发它们的onChanged回调，我们可以在此回调中处理选中状态改变逻辑。下面看一个简单的例子：

class SwitchAndCheckBoxTestRoute extends StatefulWidget {  
 @override  
 \_SwitchAndCheckBoxTestRouteState createState() => new \_SwitchAndCheckBoxTestRouteState();  
}  
  
class \_SwitchAndCheckBoxTestRouteState extends State<SwitchAndCheckBoxTestRoute> {  
 bool \_switchSelected=true; //维护单选开关状态  
 bool \_checkboxSelected=true;//维护复选框状态  
 @override  
 Widget build(BuildContext context) {  
 return Column(  
 children: <Widget>[  
 Switch(  
 value: \_switchSelected,//当前状态  
 onChanged:(value){  
 //重新构建页面   
 setState(() {  
 \_switchSelected=value;  
 });  
 },  
 ),  
 Checkbox(  
 value: \_checkboxSelected,  
 activeColor: Colors.red, //选中时的颜色  
 onChanged:(value){  
 setState(() {  
 \_checkboxSelected=value;  
 });  
 } ,  
 )  
 ],  
 );  
 }  
}

上面代码中，由于需要维护Switch和Checkbox的选中状态，所以SwitchAndCheckBoxTestRoute继承自StatefulWidget 。在其build方法中分别构建了一个Switch和Checkbox，初始状态都为选中状态，当用户点击时，会将状态置反，然后回调用setState()通知Flutter framework重新构建UI。

图3-23

### 属性及外观

Switch和Checkbox属性比较简单，读者可以查看API文档，它们都有一个activeColor属性，用于设置激活态的颜色。至于大小，到目前为止，Checkbox的大小是固定的，无法自定义，而Switch只能定义宽度，高度也是固定的。值得一提的是Checkbox有一个属性tristate ，表示是否为三态，其默认值为false ，这时Checkbox有两种状态即“选中”和“不选中”，对应的value值为true和false 。如果tristate值为true时，value的值会增加一个状态null，读者可以自行了解。

### 总结

通过Switch和Checkbox我们可以看到，虽然它们本身是与状态（是否选中）关联的，但它们却不是自己来维护状态，而是需要父组件来管理状态，然后当用户点击时，再通过事件通知给父组件，这样是合理的，因为Switch和Checkbox是否选中本就和用户数据关联，而这些用户数据也不可能是它们的私有状态。我们在自定义组件时也应该思考一下哪种状态的管理方式最为合理。