

Flutter 中的表单

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

| | |
|--|---|
| 一、Flutter 常用表单介绍..... | 1 |
| 二、TextField 文本框组件..... | 1 |
| 三、Checkbox、CheckboxListTile 多选框组件..... | 2 |
| 四、Radio、RadioListTile 单选按钮组件..... | 4 |
| 五、开关 Switch..... | 6 |

一、Flutter 常用表单介绍

Flutter 中常见的表单有 TextField 单行文本框，TextField 多行文本框、CheckBox、Radio、Switch、CheckboxListTile、RadioListTile、SwitchListTile、Slide。

二、TextField 文本框组件

TextField 表单常见属性：

| 属性 | 描述 |
|-------------|---|
| maxLines | 设置此参数可以把文本框改为多行文本框 |
| onChanged | 文本框改变的时候触发的事件 |
| decoration | hintText 类似 html 中的 placeholder border 配置文本框边框 OutlineInputBorder 配合使用 labelText label 的名称 labelStyle 配置 label 的样式 |
| obscureText | 把文本框框改为密码框 |
| controller | controller 结合 TextEditingController() 可以配置表单默认显示的内容 |

```
TextField(  
    maxLines: 10,  
    // obscureText: true,  
    decoration: InputDecoration(  
        hintText: "密码框",  
        border: OutlineInputBorder()  
    )  
)
```

```
String _username=TextEditingController();  
  
void initState() {  
    // TODO: implement initState  
    super.initState();  
    _username.text='这是文本框初始值';  
}  
TextField(  
    controller: _username,  
    onChanged: (value){  
        // print(value);  
        setState(() {  
            this._username.text=value;  
        });  
    },  
    decoration: InputDecoration(  
        hintText: "请输入您的内容",  
    ),  
)
```

三、Checkbox、CheckboxListTile 多选框组件

Checkbox 常见属性:

| 属性 | 描述 |
|-------------|---------------|
| value | true 或者 false |
| onChanged | 改变的时候触发的事件 |
| activeColor | 选中的颜色、背景颜色 |

| | |
|------------|------------------------|
| checkColor | 选中的颜色、Checkbox 里面对号的颜色 |
|------------|------------------------|

CheckboxListTile 常见属性:

| 属性 | 描述 |
|-------------|-----------------|
| value | true 或者 false |
| onChanged | 改变的时候触发的事件 |
| activeColor | 选中的颜色、背景颜色 |
| title | 标题 |
| subtitle | 二级标题 |
| secondary | 配置图标或者图片 |
| selected | 选中的时候文字颜色是否跟着改变 |

```
Checkbox(  
    value: _isSelected,  
    onChanged: (v) {  
        // print(v);  
        setState(() {  
            this._isSelected=v;  
        });  
    },  
    activeColor: Colors.red,  
    checkColor:Colors.blue  
)
```

```
CheckboxListTile(  
    value: _isSelected,  
    title: Text("nodejs 视频教程"),  
    subtitle: Text("egg.js 视频教程"),  
    onChanged: (v) {  
        setState(() {  
            this._isSelected=v;  
        });  
    },  
    activeColor: Colors.red,  
    secondary:  
    Image.network("https://www.itying.com/images/flutter/1.png"),  
    selected:_isSelected  
)
```

四、Radio、RadioListTile 单选按钮组件

Radio 常用属性:

| 属性 | 描述 |
|-------------|------------|
| value | 单选的值 |
| onChanged | 改变时触发 |
| activeColor | 选中的颜色、背景颜色 |
| groupValue | 选择组的值 |

RadioListTile 常用属性:

| 属性 | 描述 |
|-------------|---------------|
| value | true 或者 false |
| onChanged | 改变的时候触发的事件 |
| activeColor | 选中的颜色、背景颜色 |
| title | 标题 |
| subtitle | 二级标题 |
| secondary | 配置图标或者图片 |
| groupValue | 选择组的值 |

```
int _groupValue=1;
Radio(
    value: 0,
    onChanged: (v) {
      setState(() {
        this._groupValue=v;
      });
    },
    activeColor: Colors.red,
    groupValue:_groupValue ,
),
Radio(
    value: 1,
    onChanged: (v) {
      setState(() {
```

```
        this._groupValue=v;
    });
},
    activeColor: Colors.red,
    groupValue:_groupValue ,
)
```

```
int _groupValue=1;

_handelChange(v) {
    setState(() {
        _groupValue=v;
    });
}

RadioListTile(
    value: 1,
    title: Text("nodejs 视频教程"),
    subtitle: Text("egg.js 视频教程"),
    secondary:
Image.network("https://www.itying.com/images/flutter/1.png"),

    groupValue:_groupValue ,
    onChanged: _handelChange,

),
Divider(),
RadioListTile(
    value: 0,
    title: Container(
        height: 60,
        child: Text('这是文本'),
        color: Colors.red,
    ),
    // subtitle: Text("egg.js 视频教程"),
    secondary:
Image.network("https://www.itying.com/images/flutter/1.png"),
```

```
        groupValue: _groupValue ,  
        onChanged: _handelChange,  
    )
```

五、开关 Switch

| 属性 | 描述 |
|-------------|------------|
| value | 单选的值 |
| onChanged | 改变时触发 |
| activeColor | 选中的颜色、背景颜色 |

六、Flutter Vscode 调试 Flutter 项目

1、Vscode 中打开 flutter 项目进行开发

2、运行 Flutter 项目

```
flutter run
```

r 键： 点击后热加载，也就算是重新加载吧。

p 键： 显示网格，这个可以很好的掌握布局情况，工作中很有用。

o 键： 切换 android 和 ios 的预览模式。

q 键： 退出调试预览模式。

3、Vscode 默认连不上第三方模拟器解决方案

cd 到对应夜神模拟器 D:\Program Files\Nox\bin 目录 然后运行

```
nox_adb.exe connect 127.0.0.1:62001
```