一、输入框

Flutter中也有类似于Android中EditText的输入组件——TextField.

1、构造方法

TextFiled中的参数都是可选参数

```
class TextField extends StatefulWidget {

 const TextField({
 Key key,
 this.controller,
 this.focusNode,
 this.decoration = const InputDecoration(),
 TextInputType keyboardType,
 this.textInputAction,
  this.textCapitalization = TextCapitalization.none,
  this.style,
  this.textAlign = TextAlign.start,
  this.textDirection,
  this.autofocus = false,
  this.obscureText = false,
  this.autocorrect = true,
  this.maxLines = 1,
  this.maxLength,
  this.maxLengthEnforced = true,
  this.onChanged,
  this.onEditingComplete,
  this.onSubmitted,
  this.inputFormatters,
  this.enabled,
  this.cursorWidth = 2.0,
  this.cursorRadius,
  this.cursorColor,
  this.keyboardAppearance,
  this.scrollPadding = const EdgeInsets.all(20.0),
  this.dragStartBehavior = DragStartBehavior.down,
  this.enableInteractiveSelection,
  this.onTap,
  this.buildCounter,
  }) : assert(textAlign != null),
```

```
35    assert(autofocus != null),
36    assert(obscureText != null),
37    assert(autocorrect != null),
38    assert(maxLengthEnforced != null),
39    assert(scrollPadding != null),
40    assert(dragStartBehavior != null),
41    assert(maxLines == null || maxLines > 0),
42    assert(maxLength == null || maxLength == TextField.noMaxLength || maxLe
ngth > 0),
43    keyboardType = keyboardType ?? (maxLines == 1 ? TextInputType.text : Te
xtInputType.multiline),
44    super(key: key);
45
46  }
```

参数名字	参数类型	意义	必选 or 可选
key	Key	Widget 的标识	可选
controller	TextEditingController	控制 TextField 的编辑，如果没有设置，会有默认值	可选
focusNode	FocusNode	用于控制TextField是否占有当前键盘的输入焦点 它是我们和键盘交互的一个handle	可选
decoration	InputDecoration	用于控制TextField的外观显示，如提示文本、背景颜色、边框等	可选
textAlign	TextAlign	文本的对齐方式	可选
textDirection	TextDirection	文字方向	可选
keyboardType	TextInputType	用于设置该输入框默认的键盘输入类型	可选
textInputAction	TextInputAction	键盘动作按钮图标(即回车键位图标)	可选
textCapitalization	TextCapitalization	定义文本的大写格式	可选
style	TextStyle	文本样式	可选
textAlign	TextAlign	文本的对齐方式	可选
textDirection	TextDirection	文字方向	可选
autofocus	bool	是否自动获取焦点 默认为false	可选
obscureText	bool	是否隐藏正在编辑的文本，如用于输入密码的场景等，文本内容会用“•”替换 默认为false	可选

autocorrect	bool	默认为true	可选
maxLines	int	显示的最大行数	可选
maxLength	int	输入框中允许的最大字符数	可选
maxLengthEnforced	bool	是否强制限制最大字符数，默认为true true：强制限制最大字符数 false：不限制最大字符数，即使设置了maxLength也不生效	可选
onChange	ValueChanged	输入框内容改变时的回调函数；注：内容改变事件也可以通过controller来监听	可选
onEditingComplete	VoidCallback	输入框输入完成时触发，但是onEditingComplete没有参数，不会返回内容	可选
onSubmitted	ValueChanged	输入框输入完成时触发，但是onSubmitted有参数，会返回内容	可选
inputFormatters	List<TextInputFormatter>	用于指定输入格式；当用户输入内容改变时，会根据指定的格式来校验。	可选
enabled	bool	输入框是否禁用 如果为false，则输入框会被禁用，禁用状态不接收输入和事件，同时显示禁用态样式（在其decoration中定义）。	可选
cursorWidth	double	自定义输入框光标宽度	可选
cursorRadius	Radius	自定义输入框光标圆角	可选
cursorColor	Color	自定义输入框光标颜色	可选
keyboardAppearance	Brightness	设置键盘的亮度模式 只能在iOS上使用，有两种：Brightness.dart和Brightness.light	可选
scrollPadding	EdgeInsets	文本框滑动时的间距	可选
dragStartBehavior	DragStartBehavior	设置确定当用户启动拖动时拖动正式开始的时间	可选
enableInteractiveSelection	bool	是否启用交互式选择 true：长按将会选中文字，并且弹出 cut/copy/paste 的菜单	可选
onTap	GestureTapCallback	TextField的点击事件	可选
buildCounter	InputCounterWidgetBuilder	生成自定义 InputDecorator.counter 小部件的回调	可选

- keyboardType

  类型为TextInputType，用于设置输入框默认的输入类型，例如数字、日期等

TextInputType的值	含义
TextInputType.text	文本输入键盘
TextInputType.multiline	多行文本，需和maxLines配合使用(设为null或大于1)
TextInputType.number	数字；会弹出数字键盘
TextInputType.phone	优化后的电话号码输入键盘；会弹出数字键盘并显示"* #"
TextInputType.datetime	优化后的日期输入键盘；Android上会显示“: -”
TextInputType.emailAddress	优化后的电子邮件地址；会显示“@ .”
TextInputType.url	优化后的url输入键盘； 会显示“/ .”

- textInputAction

  类型为TextInputAction，键盘动作按钮图标(即回车键位图标)。

- textCapitalization

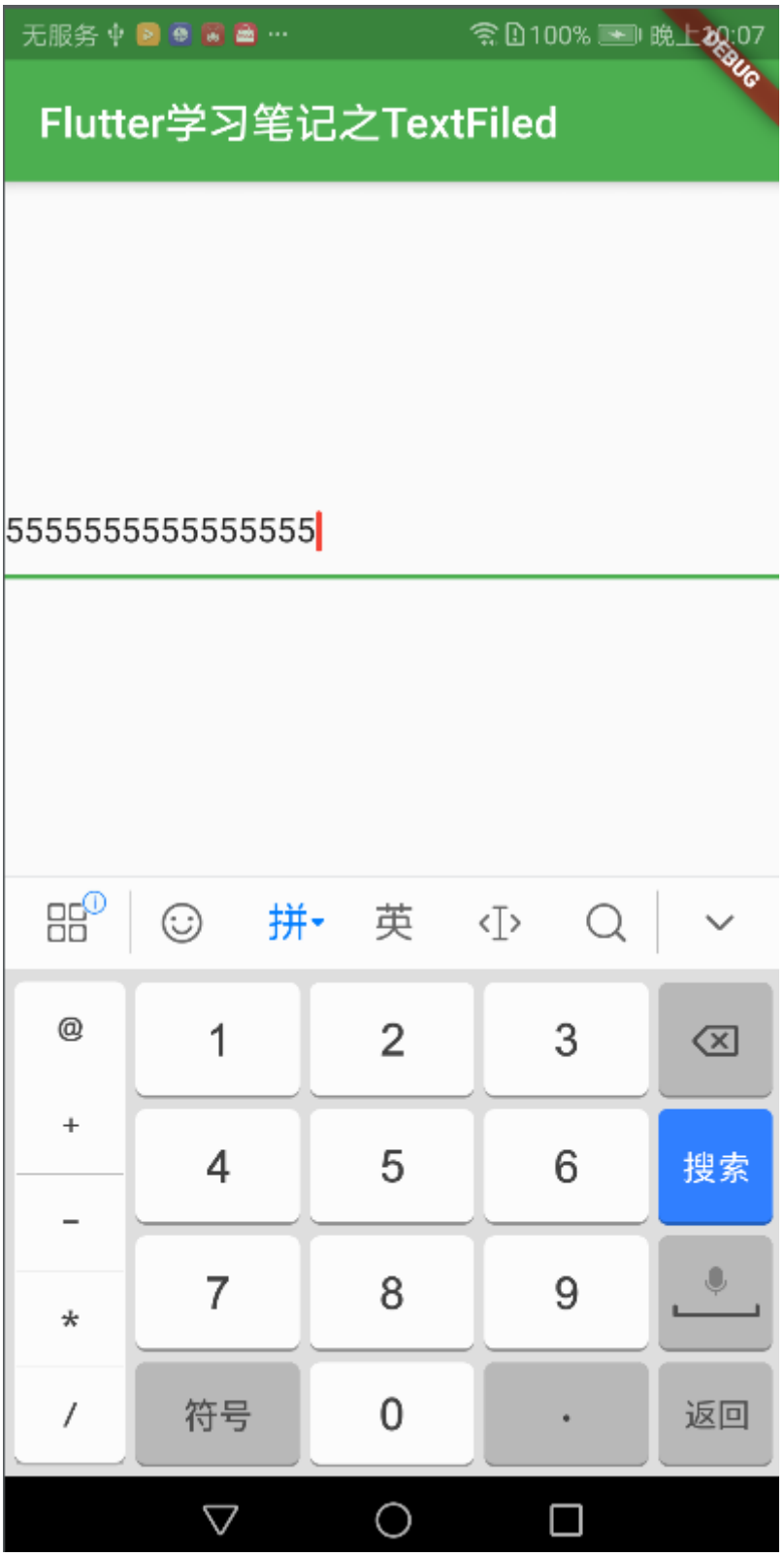
  定义文本的大写格式

TextCapitalization的值	含义
TextCapitalization.none	全部小写
TextCapitalization.words	每个单词的首字母大写
TextCapitalization.sentences	每个句子的首字母大写
TextCapitalization.characters	每个字每大写

2、代码范例

```
class CustomTextFieldWidget extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
 // TODO: implement build
 return MaterialApp(
 title: "Flutter App",
 theme: ThemeData(primaryColor: Colors.green),
 home: Scaffold(
 appBar: AppBar(
  title: Text("Flutter学习笔记之TextFiled"),
  ),
  body: Center(
```

```
  child: TextField(
  onChanged: (String data) {
  print(data);
  },
  decoration: InputDecoration(
  hintText: "请输入标题",
  ),
  textInputAction: TextInputAction.search,
  keyboardType: TextInputType.number,
  cursorColor: Colors.red,
  cursorRadius: Radius.circular(12),
  cursorWidth: 3,
  ),
  ),
  ),
  );
  }
}
```



二、表单

1、简介

Form是将多个表单元素组合起来到的一个容器，可以将多个表单元素合并起来一起校验。

例如：登录注册或者创建群组等对用户名、密码等的校验，
Form中的表单元素时FormField以及其子类，我们常用的是以下两个

- DropdownButtonFormField
- TextFormField

使用表单的方法是：

- 创建Form，添加GlobalKey
- 向Form中添加表单元素，为每个表单元素添加校验逻辑
- 添加一个按钮去提交，并验证表单，提交并验证表单需要用Form的FormState方法

2、代码范例

```
class TestFormWidget extends StatefulWidget {
 @override
 State<StatefulWidget> createState() {
 // TODO: implement createState
 return FormWidgetState1();
 }
}


class FormWidgetState1 extends State<TestFormWidget> {
 final GlobalKey<FormState> _formKey = GlobalKey<FormState>();

 String _userGender = '男';
 String _userName;
 String _userPassword;

 @override
 Widget build(BuildContext context) {
 return MaterialApp(
 title: "Flutter Demo",
 theme: ThemeData(
 primaryColor: Colors.blue,
 ),
 home: Scaffold(
 appBar: AppBar(title: Text("Flutter UI基础Widget -- Form")),
 body: Form(
```

```
key: _formKey,
child: Column(
children: <Widget>[
DropdownButtonFormField<String>(
value: _userGender,
items: ['男', '女']
.map((label) => DropdownMenuItem(
child: Text(label),
value: label,
))
.toList(),
onChanged: (value){
setState(() {
_userGender = value;
});
},
onSaved: (value){
_userGender = value;
},
),
TextFormField(
decoration: InputDecoration(hintText: '用户名'),
validator: (value) { //
if (value?.length <= 5) {
return '用户名必须大于 5 个字符';
}
},
onSaved: (value) {
_userName = value;
},
),
TextFormField(
decoration: InputDecoration(hintText: '密码'),
obscureText: true,
autovalidate: true,
validator: (value) {
if (value?.length <= 8) {
return '密码必须大于 8 个字符';
}
},
```

```
67   onSaved: (value) {
68   _userPassword = value;
69   },
70   ),
71   RaisedButton(
72   child: Text('注册'),
73   onPressed: () {
74   if (_formKey.currentState.validate()) {
75   _formKey.currentState.save();
76   print(_userGender);
77   print(_userName);
78   print(_userPassword);
79   }
80   },
81   )
82   ],
83   ),
84   )),
85   );
86   }
87 }
```

3、截图


无服务
100%
下午3:14
DEBUG
Flutter UI基础Widget -- Form
男
啦啦啦啦啦啦啦啦
密码必须大于 8 个字符
注册