

Flutter ClipRect 使用介绍

获取失败

ClipRect 介绍

ClipRect 控件默认是通过限制子 widget 的绘制区域来达到裁剪的效果的，通过 custom clipper，可以自定义裁剪的大小跟坐标

ClipRect 的定义如下

```
ClipRect({Key key,  
          CustomClipper<Rect> clipper,  
          Clip clipBehavior: Clip.hardEdge,  
          Widget child })
```

- **clipper** : 自定义裁剪的大小跟局域
- **clipBehavior** : 裁剪方式，取值有：
 - Clip.hardEdge** (快速裁剪，牺牲质量，不支持抗锯齿)
 - antiAlias** (较慢的裁剪，支持抗锯齿，裁剪出来的边框相对圆滑)，
 - antiAliasWithSaveLayer** (裁剪效率最低最慢，通常比较少用到)

CustomClipper 介绍

ClipRect class 默认是没有任何裁剪效果的，需要通过 clipper 参数告诉 ClipRect 如何去裁剪，clipper 是个 CustomClipper 类型，CustomClipper 是个抽象接口类，我

们通过继承 CustomClipper，重写 `getClip` 方法可以定义一个裁剪区域，通过重写 `shouldReclip` 方法来告诉 ClipRect 当一个新的 clipper 被设置了是否需要更新裁剪区域，譬如开始设置的 clipper 裁剪坐标是从 (10,10) 开始的，新设置的 clipper 裁剪坐标是 (20,20)，那么 `shouldReclip` 需要返回 true 来通知 ClipRect 更新裁剪区域。

ClipRect 使用

下面这个例子从网络上加载一张图片，并且进行裁剪，裁剪坐标是 (10,10) 裁剪的宽高是图片的宽高减去 10

```
import 'package:flutter/material.dart';

/**
 * Created by nls on 2019/7/27.
 * Nothing.
 */
class ClipDemo extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(primaryColor: Colors.blue),
      home: HomeWidget(),
    );
  }
}
```

```

class HomeWidget extends StatefulWidget {

  @override
  State createState() => HomeState();
}

class HomeState extends State<HomeWidget> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('ClipDemo'),),
      body: Container(
        alignment: Alignment.center,
        child: clipRect()
      ));
  }

  Widget clipRect() {
    return ClipRect(
      clipper: _MyClipper(),
      child: Align(
        alignment: Alignment.topCenter,
        child: Image.network("http://img.redocn.com/sheying/20150213/mulanweichangcaoyuanfengjing_3951976.jpg",fit: BoxFit.fill),
      ),
    );
  }
}

class _MyClipper extends CustomClipper<Rect>{
  @override

```

```
Rect getClip(Size size) {  
    return new Rect.fromLTRB(10.0, 10.0, size.width - 10.0,  
size.height- 10.0);  
}  
  
@override  
bool shouldReclip(CustomClipper<Rect> oldClipper) {  
    return false;  
}  
}
```

效果如下

ClipDemo

BUG

