

- 巧解ViewPager滑动冲突 (2)
- Android UI 之 CheckBox (0)
- Android UI 之 TextView 的 (0)
- 如何在代码中给 Fragment (0)
- 在 TextView 显示前测量字 (0)
- ViewPager 控制滑动问题 (0)
- 竖直 (垂直) VerticalSee (0)
- Android UI 之自定义 Rad (0)

推荐文章

- *Viewpager实现底部滑动菜单，同时取消预加载
- *几种软负载均衡策略分析
- *浅谈Storm流式处理框架
- *阿里2016实习offer五面经验与总结
- *手把手教你打造一个心电图效果
View Android自定义View
- *三十分钟理解博弈论“纳什均衡”
- Nash Equilibrium

最新评论

ButterKnife使用详解
IT江湖小熊: @YaphetZhao: 谢谢，欢迎交流

ButterKnife使用详解
Yaphet赵勇: 总结的不错哦~

巧解ViewPager滑动冲突
IT江湖小熊: @qq_25046647: 就是一个关于事件分发，设置滑动区域的，控制事件是由哪里层层处理而已，这个...

巧解ViewPager滑动冲突
qq_25046647: 不是很明白呀，有没有demo呀，给我发一下，guofenghegang@qq.com 谢谢

百度基础地图Android SDK应用
IT江湖小熊: @HELLOLWE: 发你了

百度基础地图Android SDK应用
HELLOLWE:
lw0311@aliyun.com

百度基础地图Android SDK应用
HELLOLWE: 源码

百度基础地图Android SDK应用
IT江湖小熊: @xiongeliang8611: 说邮箱，发给你

百度基础地图Android SDK应用
xiongeliang861103: 积分不够，不了

二.常见使用方法

1) 由于每次都要在Activity中的onCreate绑定Activity，所以个人建议写一个 BaseActivity完成绑定，子类继承即可

注：ButterKnife.bind(this)；绑定Activity 必须在setContentView之后；

实现如下 (FragmentActivity 实现一样) :

```
[java] 书签 C 8

01. public abstract class BaseActivity extends Activity {
02.     public abstract int getContentViewById();
03.
04.     @Override
05.     protected void onCreate(Bundle savedInstanceState) {
06.         super.onCreate(savedInstanceState);
07.         setContentView(getContentViewById());
08.         ButterKnife.bind(this);
09.         initAllMembersView(savedInstanceState);
10.     }
11.
12.     protected abstract void initAllMembersView(Bundle savedInstanceState);
13.
14.     @Override
15.     protected void onDestroy() {
16.         super.onDestroy();
17.         ButterKnife.unbind(this); // 解除绑定，官方文档只对fragment做了解绑
18.     }
19. }
```

2) 绑定fragment

3) 绑定view

```
[java] 01. @Bind(R.id.hello_world)  
02. TextView mHelloWorldTextView;  
03. @Bind(R.id.app_name)  
04. TextView mAppNameTextView://v
```

4) 绑定资源

5) Adapter ViewHolder 绑定

```
[java] [ ] [ ] [ ] [ ] [ ] [ ]  
01. public class TestAdapter extends BaseAdapter {  
02.     private List<String> list;  
03.     private Context context;  
04.  
05.     public TestAdapter(Context context, List<String> list) {  
06.         this.list = list;  
07.         this.context = context;  
08.     }  
09.  
10.    @Override  
11.    public int getCount() {  
12.        return list==null ? 0 : list.size();  
13.    }  
14.  
15.    @Override  
16.    public Object getItem(int position) {  
17.        return list.get(position);  
18.    }  
19.  
20.    @Override  
21.    public long getItemId(int position) {  
22.        return position;  
23.    }  
24.  
25.    @Override  
26.    public View getView(int position, View convertView, ViewGroup parent) {  
27.        ViewHolder holder;  
28.        if (convertView == null) {  
29.            convertView = LayoutInflator.from(context).inflate(R.layout.layout_list_item, parent, false);  
30.            holder = new ViewHolder(convertView);  
31.            convertView.setTag(holder);  
32.        } else {  
33.            holder = (ViewHolder) convertView.getTag();  
34.        }  
35.        holder.textview.setText("item====" + position);  
36.        return convertView;  
37.    }  
38.  
39.    static class ViewHolder {  
40.        @Bind(R.id.hello_world)  
41.        TextView textview;  
42.  
43.        public ViewHolder(View view) {  
44.            ButterKnife.bind(this, view);  
45.        }  
46.    }  
}
```

6) 点击事件的绑定：不用声明view，不用setOnClickListener () 就可以绑定点击事件

a.直接绑定一个方法

```
[java] [ ] [ ] [ ] [ ] [ ] [ ]  
01. @OnClick(R.id.submit)  
02. public void submit(View view) {  
03.     // TODO submit data to server...  
04. }
```

b.所有监听方法的参数是可选的

```
[java] [ ] [ ] [ ] [ ] [ ] [ ]  
01. @OnClick(R.id.submit)  
02. public void submit() {  
03.     // TODO submit data to server...  
04. }
```

c.定义一个特定类型，它将自动被转换

```
[java] [ ] [ ] [ ] [ ] [ ] [ ]  
01. @OnClick(R.id.submit)  
02. public void sayHi(Button button) {  
03.     button.setText("Hello!");  
04. }
```

d.多个view统一处理同一个点击事件，很方便，避免抽方法重复调用的麻烦

```
[java] [D] [E] [C] [P]  
01. @OnClick({ R.id.door1, R.id.door2, R.id.door3 })  
02. public void pickDoor(DoorView door) {  
03.     if (door.hasPrizeBehind()) {  
04.         Toast.makeText(this, "You win!", LENGTH_SHORT).show();  
05.     } else {  
06.         Toast.makeText(this, "Try again", LENGTH_SHORT).show();  
07.     }  
08. }
```

e.自定义view可以绑定自己的监听，不指定id

```
[java] [D] [E] [C] [P]  
01. public class FancyButton extends Button {  
02.     @OnClick  
03.     public void onClick() {  
04.         // TODO do something!  
05.     }  
06. }
```

f.给EditText加addTextChangedListener (即添加多回调方法的监听的使用方法) , 利用指定回调，实现想回调的方法

```
[java] [D] [E] [C] [P]  
01. @OnTextChanged(value = R.id.mobileEditText, callback = OnTextChanged.Callback.BEFORE_TEXT_CHANGED)  
02. void beforeTextChanged(CharSequence s, int start, int count, int after) {  
03. }  
04.  
05. @OnTextChanged(value = R.id.mobileEditText, callback = OnTextChanged.Callback.TEXT_CHANGED)  
06. void onTextChanged(CharSequence s, int start, int before, int count) {  
07. }  
08.  
09. @OnTextChanged(value = R.id.mobileEditText, callback = OnTextChanged.Callback.AFTER_TEXT_CHANGED)  
10. void afterTextChanged(  
11. }  
12.
```

7)对一组View进行统一操作

a.装入一个list

```
[java] [D] [E] [C] [P]  
01. @Bind({ R.id.first_name, R.id.middle_name, R.id.last_name })  
02. List<EditText> nameViews;
```

b.统一操作处理，例如设置是否可点，属性等

```
[java] [D] [E] [C] [P]  
01. ButterKnife.apply(nameViews, DISABLED);  
02. ButterKnife.apply(nameViews, ENABLED, false);
```

8)可选绑定：默认情况下，“绑定”和“监听”绑定都是必需的。如果不能找到目标视图，则将抛出异常。所以做空处理

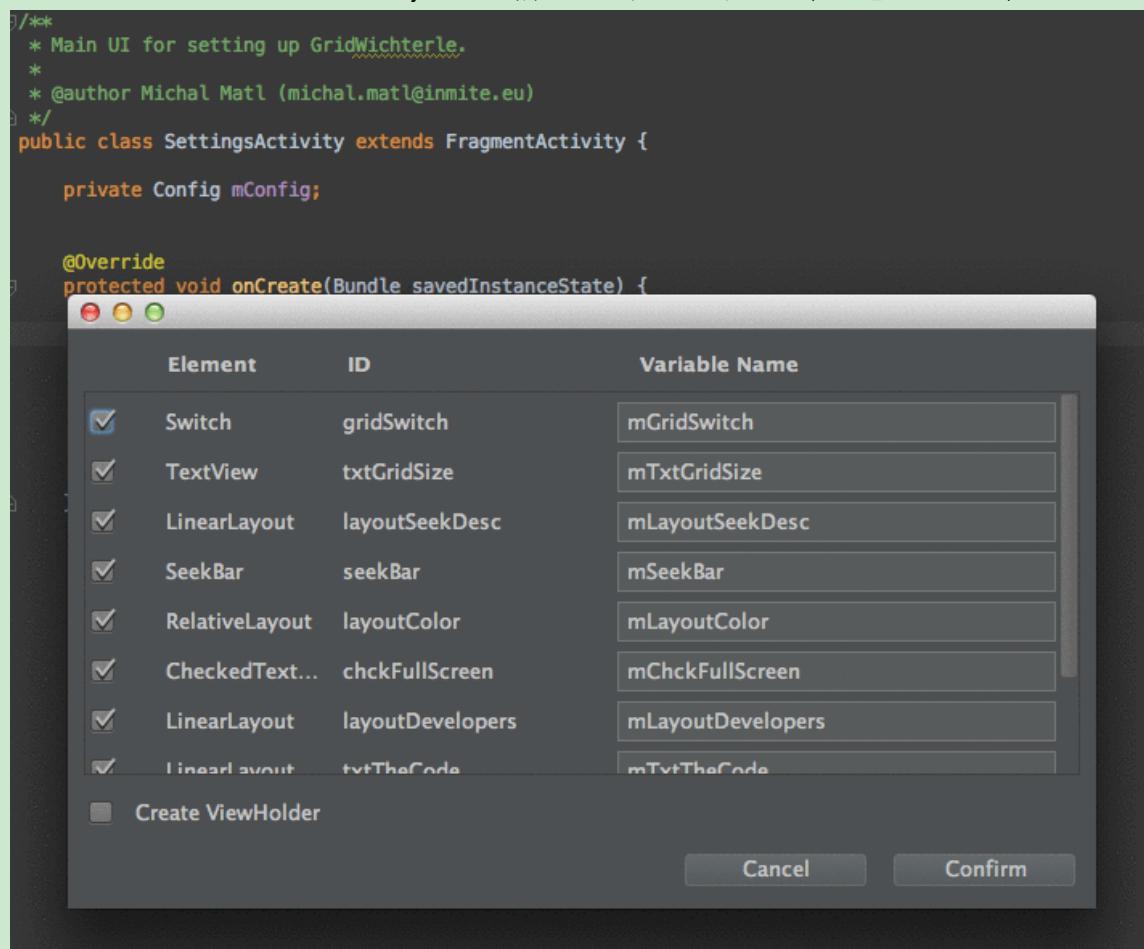
三、

```
关闭 [X]  
01. TextView mightNotBeThere;  
02.  
03. void onMaybeMissingClicked() {  
04.  
05.  
06.     -keep class **$ViewBinder { *; }  
07.     -keepclasseswithmembernames class * {  
08.         @butterknife.* <fields>;  
09.     }
```

```
08.  
09.     -keepclasseswithmembernames class * {  
10.         @butterknife.* <methods>;  
11.    }
```

四、Zelezny插件的使用

在AndroidStudio->File->Settings->Plugins->搜索Zelezny下载添加就行，可以快速生成对应组件的实例对象，不用手写。使用时，在要导入注解的Activity或Fragment或ViewHolder的layout资源代码上，右键——>Generate——Generate ButterKnife Injections，然后就出现如图的选择框。(此动态图来自官网)



顶
4 | 踩
0

上一篇 windows git 环境搭建

下一篇 Android UI 之自定义RadarView——高仿微信雷达扫描

我的同类文章

android (12)

- Android UI 之自定义Radar... 2015-12-07 阅读 328
- Android使用命令生成签名... 2015-09-10 阅读 243
- 巧解ViewPager滑动冲突 2015-04-08 阅读 712
- Android UI 之TextView的妙用 2015-04-01 阅读 177
- 如何在代码中给Fragment... 2014-12-17 阅读 209
- Android UI 之ToggleTextVi... 2015-09-11 阅读 94
- 一种夜间模式简单实现方案 2015-05-21 阅读 255
- Android UI 之CheckBox的... 2015-04-06 阅读 472
- 百度基础地图Android SDK... 2015-01-29 阅读 546

更多文章

猜你在找

■ Android底层技术：Java层系统服务(Android Service)

- Android开发—Nfc
- Oracle、SQL、PLSQL 编程从入门到精通
- SpringMVC框架（基于spring3.0）
- SpringMVC Spring Hibernate框架整合视频

查看评论

1楼 Yaphet赵勇 2016-03-14 18:03发表



总结的不错哦~

Re: IT江湖小熊 6天前 14:31发表



回复Yaphet赵勇：谢谢，欢迎交流

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved