



杨振威

案例

Android开发中的一些小技巧

目录

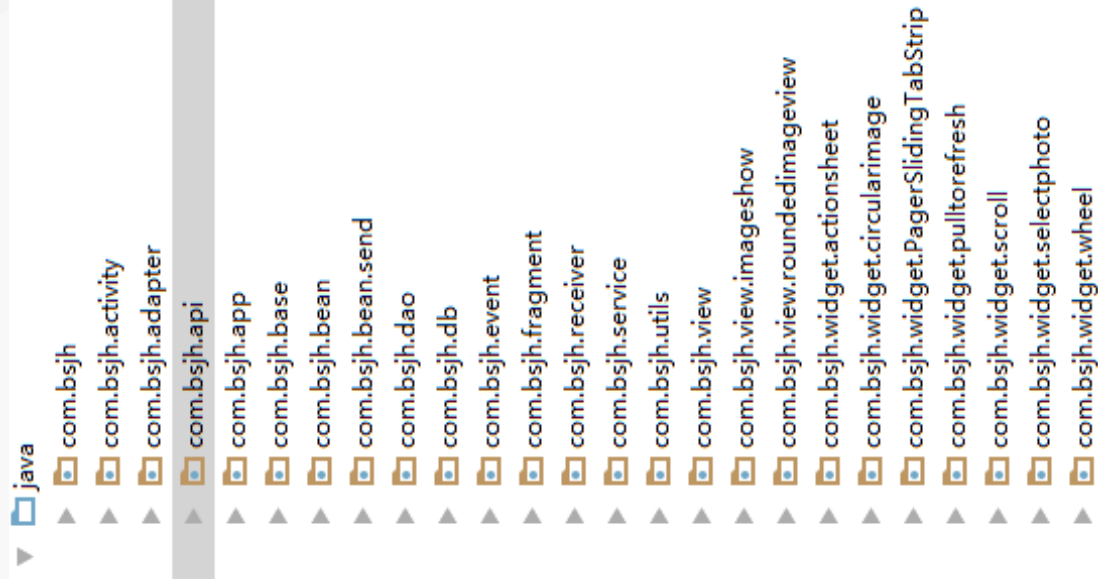
PAGE DIRECTORY

- 1、优化项目结构
- 2、定义全局exception
- 3、定义一些基类
- 4、实体化编程
- 5、Adapter模板
- 6、Fragment中的上下文
- 7、命名技巧
- 8、布局优化
- 9、android studio常用插件
- 10、全局的exception
- 11、Handler+Runnable的妙用
- 12、书籍推荐和公众号推荐

1、优化项目结构

现在的项目框架有很多种，比如

MVC，MVP，MVVM等，但是如何能让你的项目结构清晰明了，让你的团队成员以最小的学习成本尽快的融入到项目开发中，我是这样规划项目结构目录的：



2、定义全局application

定义全局application，用于初始化、保存和调用一些全局的应用配置。

```
// 注册App异常崩溃处理器
// Thread.setDefaultUncaughtExceptionHandler (AppException.getAppExceptionHandler(this));
super.onCreate();
//注册初始化OkHttp
OkHttpClient okHttpClient = new OkHttpClient.Builder()
    .addInterceptor(new LoggerInterceptor("TAG"))
    .connectTimeout(10000L, TimeUnit.MILLISECONDS)
    .readTimeout(10000L, TimeUnit.MILLISECONDS)
    //其他配置
    .build();

OkHttpUtils.initClient(okHttpClient);

this.hostConfig = getHostConfigFromXml();
initImageLoader(getApplicationContext());
mApplication = this;

SDKInitializer.initialize(getApplicationContext());

/** 百度定位
    mLocationClient = new LocationClient(getApplicationContext()); // 声明LocationClient类
    initLocation();
    mLocationClient.registerLocationListener(myListener); // 注册监听函数
```



```
//jpush init
JPushInterface.setDebugMode(true); // 设置开启日志,发布时请关闭日志
JPushInterface.init(this); // 初始化 JPush

// 初始化轨迹服务客户端
client = new LBSTraceClient(mAppApplication);

// 初始化轨迹服务
trace = new Trace(mAppApplication, serviceId, entityId, traceType);

// 设置定位模式
client.setLocationMode(com.baidu.trace.LocationMode.High_Accuracy);

mHandler = new TrackHandler(mAppApplication);
```

3、定义一些基类

定义一些常用的基类，在基类里定义一些全局变量、方法以及初始化一些数据等。

```
▼ com.bsjh.base
  BaseActivity
  BaseBean
  BaseFragment
  BaseLayout
  BaseView
  MyBaseAdapter
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    imageLoader = ImageLoader.getInstance();
    context = this;
    gson = new Gson();

    // 添加Activity到堆栈
    AppManager.getAppManager().addActivity(this);

    initVariables();
    initView(savedInstanceState);
    loadData();
}
```

```
// 返回动画
public void backAnim() { overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_right); }

// 前进动画
public void intoAnim() { overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left); }
```

4、实体化编程

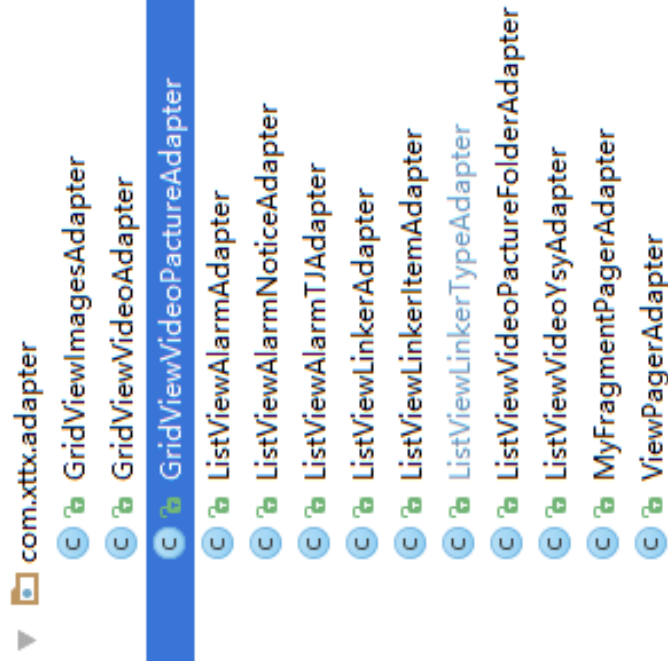
把api发送和返回的JSON数据，转换成实体类，方便取值赋值操作。



```
showProgressDialog(context, "加载中,请稍候...");
TokenSend tokenSend=new TokenSend("main",AppContext.getApp().getToken());
OkHttpUtils
    .get()
    .url(URLs.URL_API_HOST)
    .params(DataTools.beanToMap(new Gson(), tokenSend))
    .build()
    .execute(new StringCallback() {
        @Override
        public void onError(Call call, Exception e, int id) {
            AppException2.network(e, context);
            closeProgressDialog();
        }
    })
    .@Override
    public void onResponse(String response, int id) {
        Log.i("TAG", "response="+response);
        WalkInfo walkInfo=new Gson().fromJson(response, WalkInfo.class);
        if (!TextUtils.isEmpty(walkInfo.getCAPACITY()) {
            tvActivityWalkCapacity.setText(walkInfo.getCAPACITY() + "%");
        } else {
            tvActivityWalkCapacity.setText("0%");
        }
        if (!TextUtils.isEmpty(walkInfo.getTIMEKEYCASH())) {
            tvActivityWalkMonthcount.setText(walkInfo.getTIMEKEYCASH());
        }
    }
}
```


5、adapter模板

很多人喜欢把adapter嵌入到activity中，我更喜欢把adapter独立出来，并且统一编码风格，这样有问题很容易被发现排除。



```
▼ com.xtx.adapter
  ◉ GridViewImagesAdapter
  ◉ GridViewVideoAdapter
  ◉ GridViewVideoPactureAdapter
  ◉ ListViewAlarmAdapter
  ◉ ListViewAlarmNoticeAdapter
  ◉ ListViewAlarmTJAdapter
  ◉ ListViewLinkerAdapter
  ◉ ListViewLinkerItemAdapter
  ◉ ListViewLinkerTypeAdapter
  ◉ ListViewVideoPactureFolderAdapter
  ◉ ListViewVideoYsyAdapter
  ◉ MyFragmentPagerAdapter
  ◉ ViewPagerAdapter
```

6、Fragment中的上下文

一般我们在Fragment中获得上下文喜欢用getActivity(),但我更喜欢另一种方法。

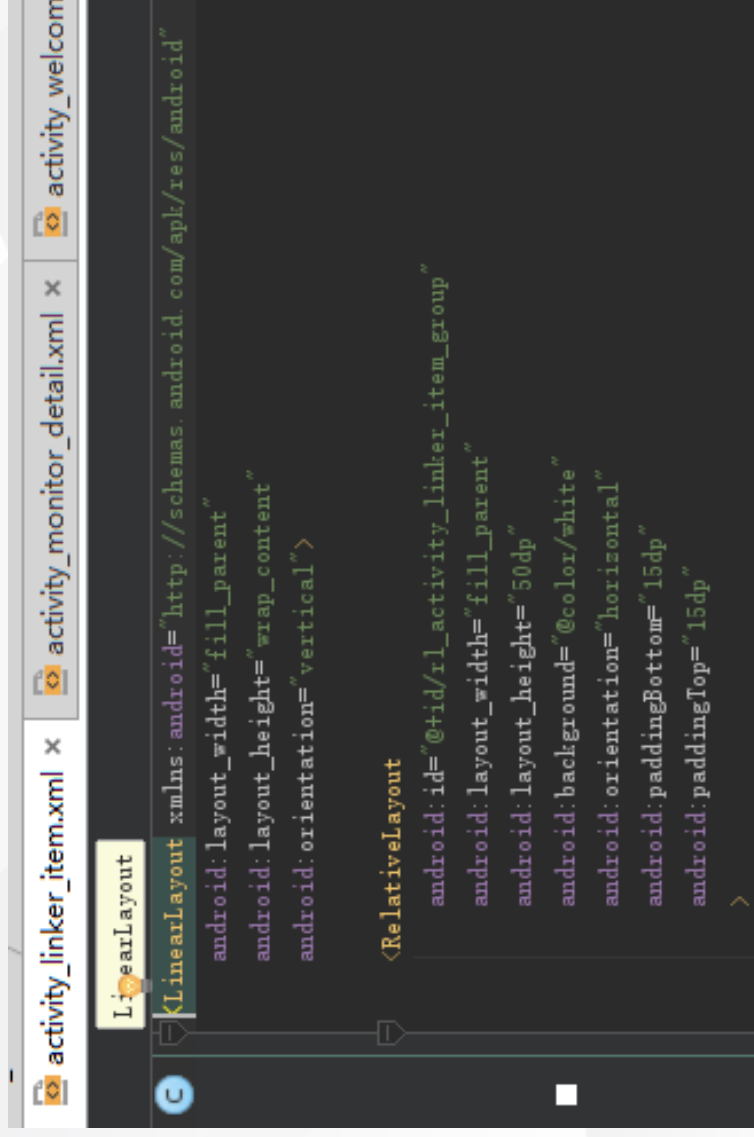
```
@Override
public void onAttach(Activity activity) {
    // TODO Auto-generated method stub
    this.activity = activity;
    super.onAttach(activity);
}
```

```
SharedPreferences sef = activity.getSharedPreferences(
    "isRememberPassword", Context.MODE_PRIVATE);
```



7、layout中的控件命名技巧

控件缩写_layout名字_功能



8、布局优化

结合代码讲解。

- 1、<include/>
- 2、减少嵌套
- 3、< merge/>
- 4、用TextView同时显示图片和文字
- 5、灵活使用TextView的行间距
- 6、用LinearLayout自带的分割线
- 7、灵活使用Space控件



9、android studio常用插件

- 1、ButterKnife Zelezny
- 2、GsonFormat



10、全局exception

在全局的exception中，把常见的一些exception捕捉到，并且自动弹出toast提示。这样在activity中，我们只需要把异常抛出即可被自动捕捉。结合代码讲解。



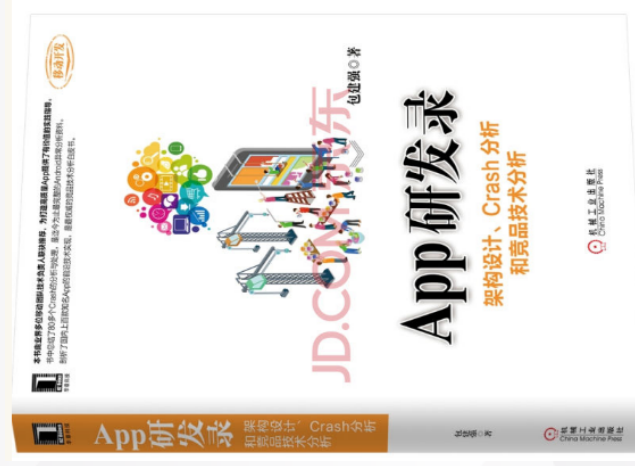
11、Handler+runnable的妙用

这是我在一个温控项目中解决问题时候的突然灵光一现，觉得这种思路不错，这种思路在我以后的项目中也常用到。传递一种思路，结合项目代码讲解。



12、书籍和公众号推荐

推荐两本书和我认为不错的两个公众号



两个微信公众号
1、鸿洋
2、移动开发前线



THANK YOU —