# Rxjava+Retrofit项目使用文档

### 简介

RxjavaRetrofitProject，字如其名，这个项目是Rxjava2+Retrofit2为主网络框架的项目，未使用MVP模式，响应数据使用String类型避免非json响应数据错误，后续会陆续加入MVP模式，以及Gson解析响应，上传下载功能。

\*此项目仅是个人集成网络框架项目，不喜勿喷。下面直接说一下，该框架使用的一些知识点，以及使用说明。

### 技术点

1：Retrofit2

retrofit2.0后：

BaseUrl要以/结尾；

@GET 等请求不要以/开头；

@Url: 可以定义完整url，不要以 / 开头。

Retrofit实例化：

Retrofit retrofit = **new** Retrofit.Builder()  
 .baseUrl(**"http://temp.jiangxunlu.com/api/selan/api/web/"**)

// Converter是Retrofit为我们提供用于将ResponseBody转换为我们想要的类型（这里我们转化为String类型），也可以添加GsonConverterFactory，转化为我们需要的bean  
.addConverterFactory(StringConverterFactory.*create*())

// CallAdapter 为我们提供Rxjava支持，我们可以使用Observable代替Call，通过RxJavaCallAdapterFactory为Retrofit添加RxJava支持  
.addCallAdapterFactory(RxJava2CallAdapterFactory.*create*())

// 添加OkhttpClient  
 .client(*createOkHttpClient*())  
 .build();

2：Rxjava2

通过subscribe连接observable（被观察者/顾名思义，观察得到的结果）和observer（观察者），即observable.subscribe(observer).

自定义DialogObserver实现Observer<T>,实现其中onSubscribe，onNext,onError,onComplete方法，根据响应数据，做具体处理。

3：Gson

4：Interceptor

### GitHub地址

https://github.com/onlyWYY/RX\_Retrofit.git

### 集成

项目集成主要是gradle添加依赖

*/\*Retrofit2\*/*compile **'com.squareup.retrofit2:retrofit:2.3.0'**compile **'com.squareup.retrofit2:converter-gson:2.3.0'**compile **'com.squareup.retrofit2:adapter-rxjava2:2.3.0'***/\*Rxjava2\*/*compile **'io.reactivex.rxjava2:rxjava:2.1.3'**compile **'io.reactivex.rxjava2:rxandroid:2.0.1'***/\*log\*/*implementation **'com.squareup.okhttp3:logging-interceptor:3.9.1'**compile(**'com.github.ihsanbal:LoggingInterceptor:2.0.5'**) {  
 exclude **group**: **'org.json'**, **module**: **'json'**}

### 使用

先看实例：

RequestCommand.*requestConfig*(**new** configObserver(MainActivity.**this**));

**private class** configObserver **extends** DialogObserver<String> {  
 **public** configObserver(Context mContext) {  
 **super**(mContext);  
 }  
  
 @Override  
 **public void** success(String info) {  
  
 }  
}

代码简洁明了，只需要实现DialogObserver对应的success(必须)，onNext,onError方法，做不同的响应处理即可。