**中山大学移动信息工程学院本科生实验报告**

**（2017年秋季学期）**

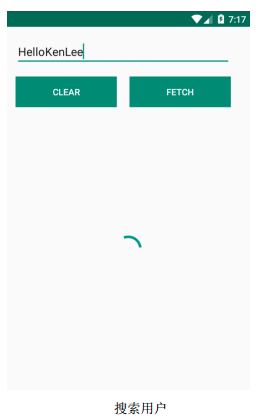
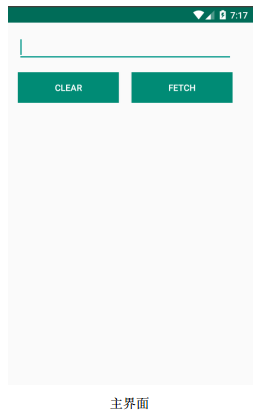
课程名称：移动应用开发 任课教师： 郑贵峰

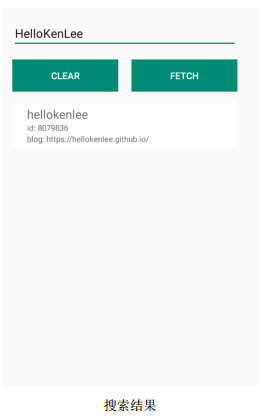
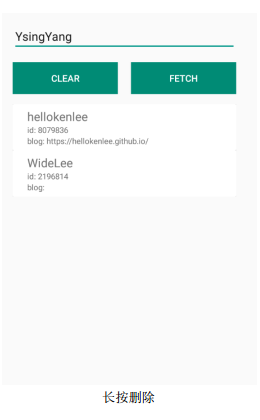
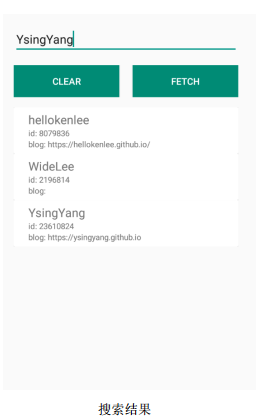
|  |  |  |  |
| --- | --- | --- | --- |
| 年级 | 2015 | 专业（方向） | 移动信息工程（M2） |
| 学号 | 15352193 | 姓名 | 梁公勋 |
| 电话 | 15066299571 | Email | 1104695389@qq.com |
| 开始日期 |  | 完成日期 |  |

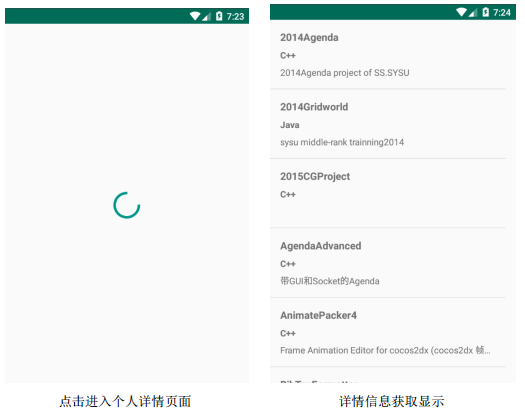
# 实验题目

Retrofit+RxJava+OkHttp 实现网络请求

# 实现内容



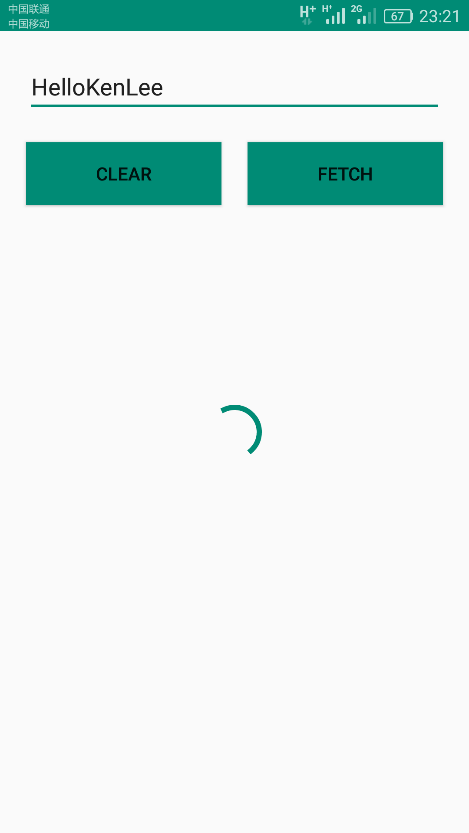
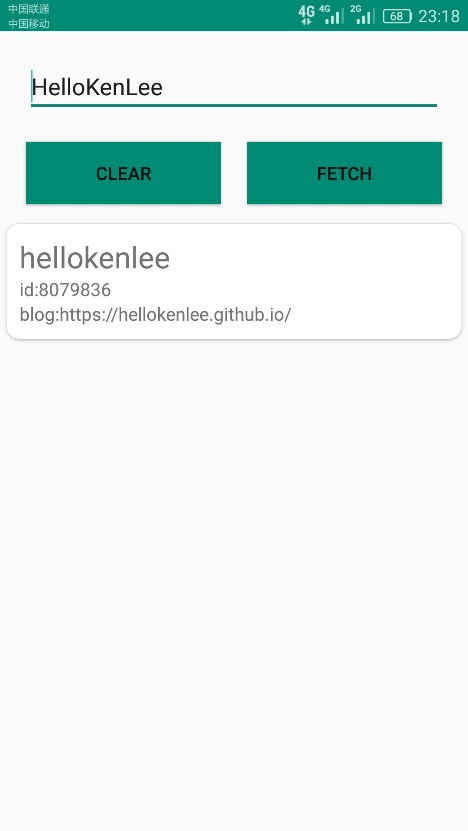
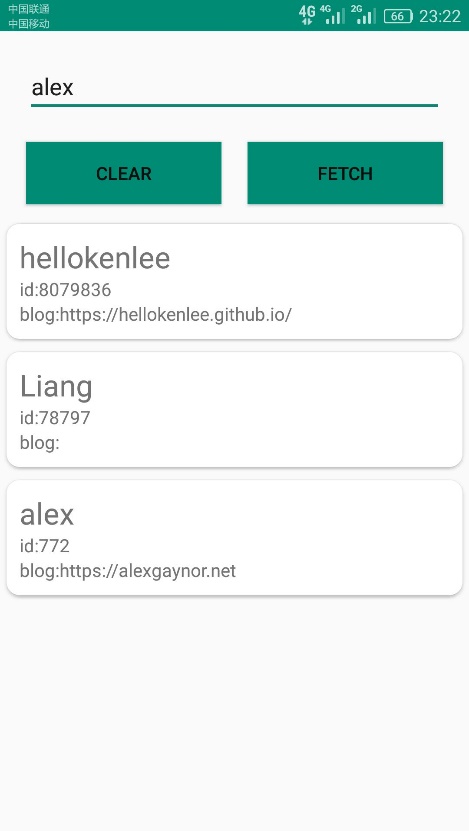


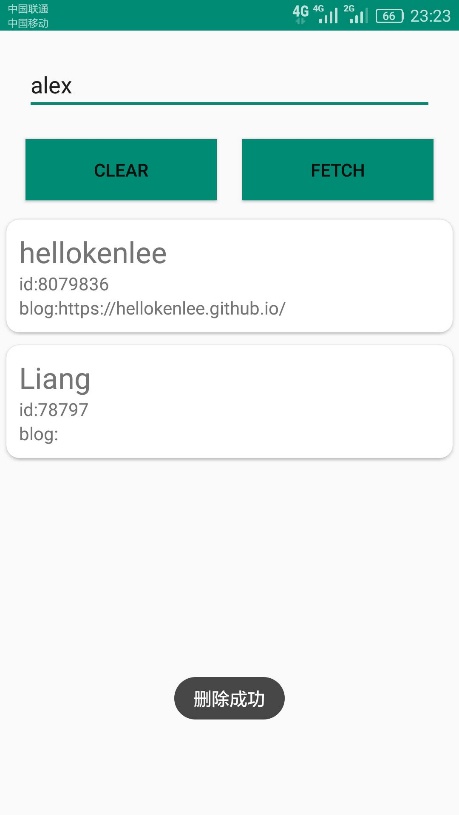
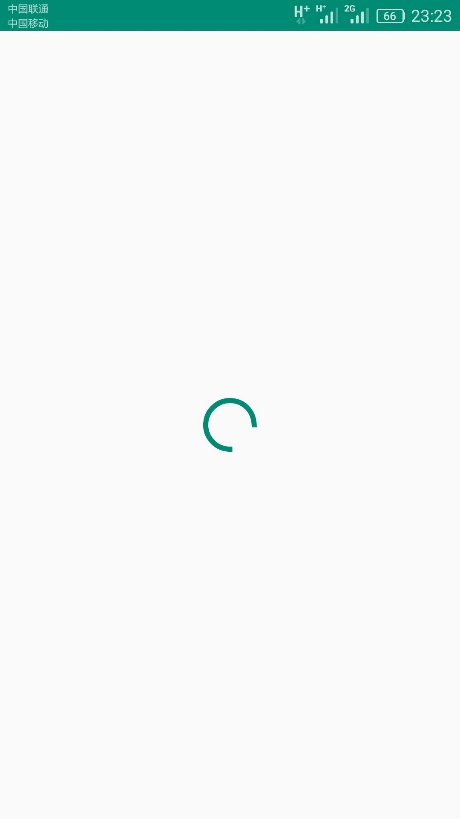
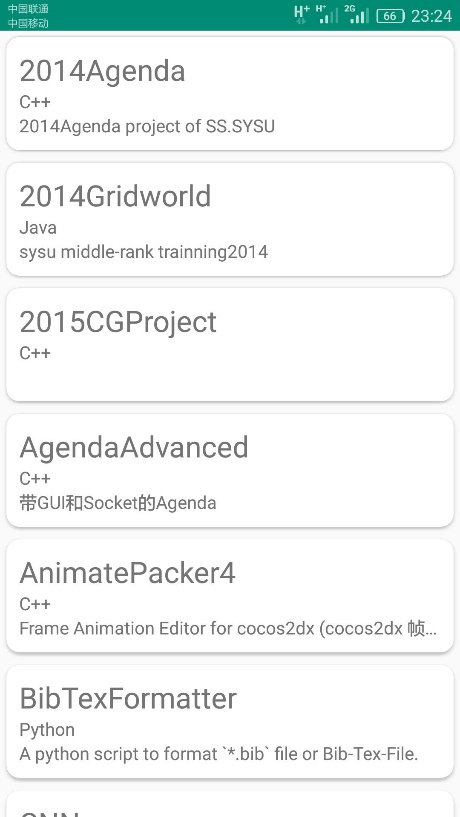
对于 User Model, 显示 id, login, blog

对于 Repository Model, 显示 name, description,language(特别注意， 如果 description 对于 1 行要用省略号代替)

# 课堂实验结果

1. 实验截图

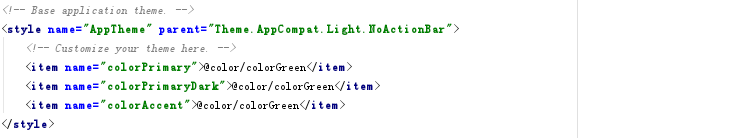
1. 实验步骤以及关键代码

**一、布局文件中的cardView等相关代码**

在这次的布局中，我们使用了CardView这一控件作为item项的布局。CardView有很多美观的属性，这里使用了cardCornerRadius来修饰圆角。使用了contentPadding修改了内边距。



另外，在这次的布局中，我们还修改了style文件，来改变APP的主题。其中使用NoActionBar删除了标题栏，修改了colorPrimaryDark的颜色改变了通知栏的颜色，修改了colorAccent改变了EditText的下划线以及ProgressBar的颜色。



为了使描述只显示单行且之后使用省略号代替，设置TextView属性如下。其中ellipsize设置在最后添加省略号，singleLine设置了单行显示。



**二、建立model类**

本次实验定义了两个model类GitHub和Repos，model类中的类型和名称一定要和API中返回的类型及名称相对应，这样才会有相应的数据返回。

GitHub model类如下所示，定义了登录名，博客网址，id，类型均为String。



Repos model类如下所示，包括姓名、使用的编程语言，描述，类型也都是String。不过我们在获取仓库数据时，得到的是一个model类的列表。



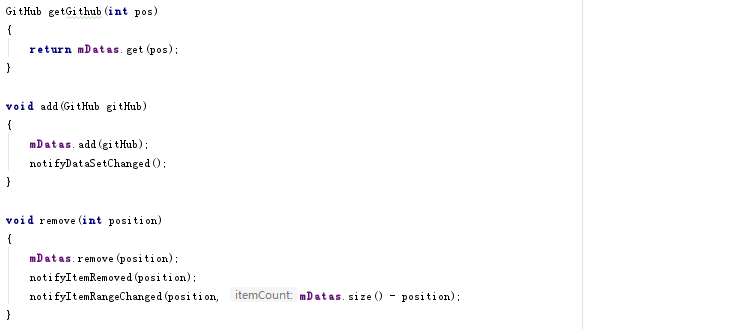
**三、自定义RecyclerView的adapter**

这一部分在lab3中是重点，这里不在解释绑定及自定义监听器等，只是列一下自定义的后边会使用的几个方法。

函数getGithub根据下标返回适配器列表中相应的GitHub对象。主要是因为这次在MainActivity中没有用来存储RecyclerView中要展示项的列表，只存在于适配器中，所以需要这个方法。

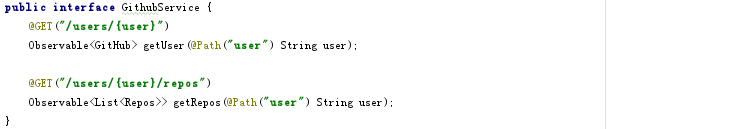
函数add是向适配器数据列表中添加项，先是向列表中添加了项，然后使用notifyDataSetChanged方法调整。因为我们每次通过网络抓取数据之后都要在RecyclerView中显示，所以需要此函数。

函数remove根据下标，删除RecyclerView中相应的项。先是从列表中删除，然后使用notifyItemRemoved方法从RecyclerView中移除，最后还要使用notifyDataSetChanged方法调整RecyclerView的项。



**四、接口的实现**

我们需要给 retrofit 对象提供相应的 Interface。声明了一个接口类，这个类中有两个方法。一个是用来获取用户的先关信息的，返回类型是GitHub model类，另一个是获取相关仓库的信息，返回类型是Repos model类的列表。Observable 为 RxJava 中的类型。这里只用到了GET方法，括号里的为baseurl后所加上的用来访问的url。从 API 获取到的是一个 JSON 格式的数据， 之后需要使用到 GsonConverter 将其“转换”为所需要的 model。



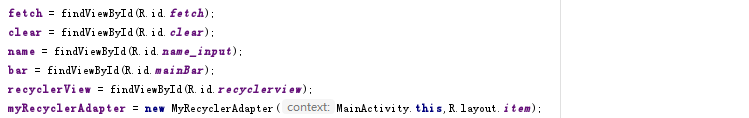
**五、MainActivity中的相关代码**

在MainActivity中主要需要实现抓取账户相关信息的部分。

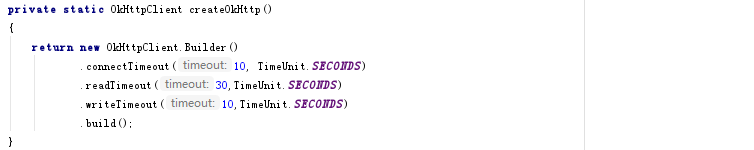
首先声明了以下变量。BASEURL是我们所要访问的网站API的基地址。其余的都是控件，最后一个是自定义的RecyclerView的适配器。



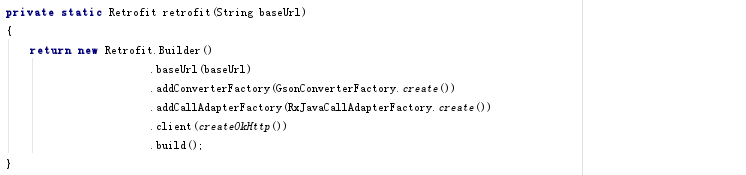
变量的初始化如下所示。主要是控件与布局文件中的绑定，然后新建了一个adapter。



接着是配置OkHttpClient对象。声明了一个静态函数，然后使用Builder创建一个OkHttpClient对象并将其返回。使用connectTimeout方法规定了连接超时的时间，使用readTimeout方法规定了读超时的时间，使用了writeTimeout方法规定了写超时的时间。



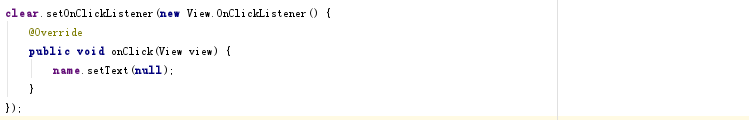
接下来是构造Retrofit对象。同样也是声明了一个静态函数，使用Builder方法创建一个Retrofit对象将其返回。使用baseUrl传入了我们所要访问的网络API的基地址。使用addConverterFactory方法将返回的JSON类型转换为我们所定义的model类。使用addCallAdapterFactory配合RxJava的使用。使用了client配置了OKHttpClient。



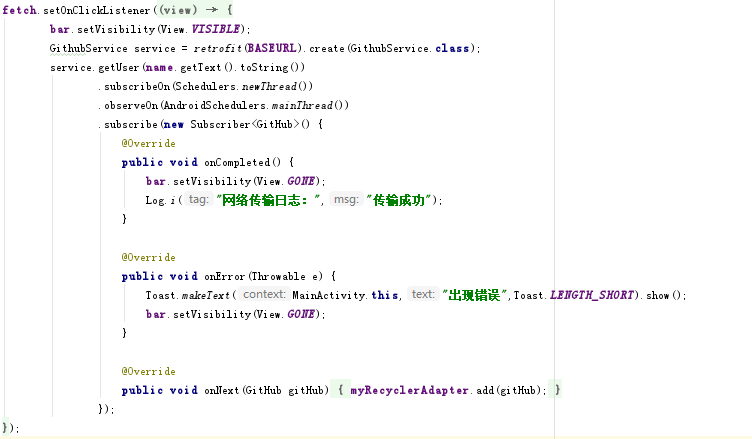
在onCreate函数中，我们首先为RecyclerView设置了布局管理器，然后设置了适配器，并将ProgressBar设置为不可见。



然后是一系列的监听事件。Clear按钮监听到单击事件后，就将输入框给清空。

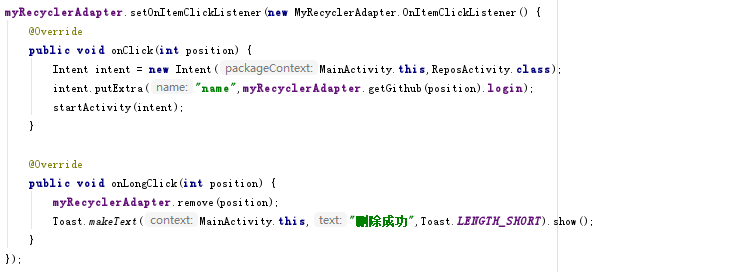


Fetch按钮监听到单击事件后，首先将ProgressBar设置为可见。然后通过retrofit创建一个访问实例，并对这个访问实例进行相应配置。



使用getUser调用了接口中的方法，获取了返回的model类。使用subscribeOn新建一个线程处理网络访问，使用observeOn规定在主线程处理返回结果。使用Subscriber建立订阅关系。然后重载3个方法。OnCompleted方法是传输完成后调用的，这里将ProgressBar设置为不可见，输出日志提示传输成功。OnError方法是传输失败时所调用的，这里弹出一条Toast提示出现错误，也设置ProgressBar为不可见。OnNext方法是处理访问结果的函数，我们将得到的model类使用自定义适配器的add方法添加到RecyclerView中显示出来。

接下来是RecyclerView中每一项的监听事件。监听到长按事件后就调用自定义适配器中的remove方法将其从RecyclerView中移除并弹出Toast信息提示删除成功。监听到单击事件后，我们需要跳转到这个用户的仓库界面。使用intent设置好要跳转的activity，然后将这一项的用户名传过去，以供另一个activity进行网络访问时使用。最后使用startActivity进行跳转。



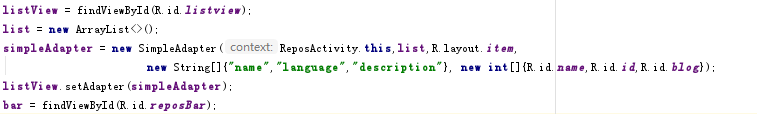
**六、仓库显示的ReposActivity的相关代码。**

这个activity主要是显示某一个用户在GitHub中的仓库。

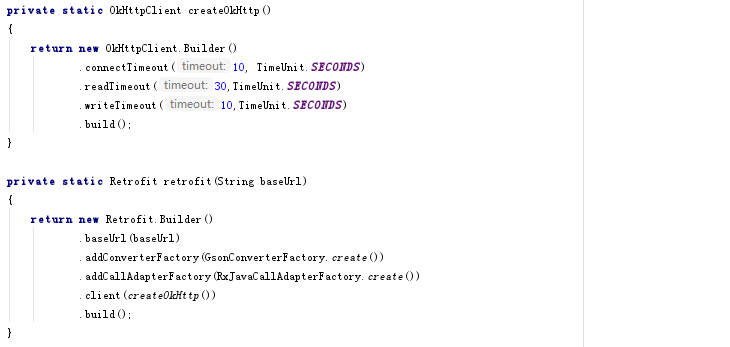
这个activity的参数如下所示。BASEURL还是基地址，listview用来展示获取到的仓库。还有listview所要使用的适配器simpleAdapter。最后是向simpleAdapter中所要传入的存储map的list。



这些变量的初始化如下所示。控件对象与布局中的控件绑定。然后list新建一个ArrayList。适配器新建一个simpleAdapter并传入context，数据列表list，每一项要绑定的布局。以及数据与布局中控件的对应关系。最后为listview设置适配器。



另外，由于这个activity也要进行网络访问，所以也需要建立Retrofit和OKHttpClient对象，方法同MainActivity中完全一样，访问网络的API地址也一样，这里不再解释。



然后是在onCreate函数中访问网络获取数据。首先也是通过Retrofit对象创建一个接口实例，然后对这个实例进行配置。除了调用接口中的方法不同，其余和之前的都一样。这里调用接口返回的数据类型是一个存储Repos model类的列表list。在OnCompleted中，将ProgressBar设置为不可见，输出日志提示传输成功。在OnError方法中弹出一条Toast提示出现错误。在OnNext方法中，我们便利这个list，然后对于其中的每一个Repos类，我们都新建一个map，然后将其数据以键值对的方式一一存储到这个新建的map中去，注意这里的key要和之前新建simpleAdapter时传入的key相一致。然后我们将这个map添加到与适配器绑定的list中。所有得到的数据都按此种方法处理完成后，调用适配器的notifyDataSetChanged方法调整列表项并将其显示出来。



1. 实验遇到困难以及解决思路

1、实现接口的时候报错

这个主要是自己以为，在代码里写了retrofit2和rx，然后IDE报错，会提醒添加依赖。结果IDE的报错里没有提醒添加依赖。弄了好久，最后自己手动添加之后，一个build之后，就都可以了。

2、使用service获取数据时报错崩溃。

这个主要是因为model类里的数据类型和API中的不对应，结果就是数据类型对应的能够获取到，而数据类型不对应的获取之后再使用就崩溃了。

# 课后实验结果

## 实验思考及感想

在这次的实验中，我们简单的实现了APP网络访问。网络访问是当前APP最基本最重要的功能之一。掌握了这一方面的知识，也为我们开发更好的APP打下了基础。另外，在这次实验中较多的用到了之前的知识，然后就明显感受到之前的知识不是很熟练了，经常要翻看之前的代码或者实验文档。果然还是多用才能记得住啊。

作业要求：

1. 命名要求: 学号\_姓名\_实验编号，例如15330000\_林XX\_lab1。
2. 实验报告提交格式为pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按0分处理。