链滴

Retrofit+Rxjava 服务器 IP 轮询重试机制实现

作者：idxiu

原文链接：https://ld246.com/article/1482287095942

来源网站：链滴

<p style="box-sizing: border-box; outline: none; line-height: 1.7em; color: rgb(51, 51, 51); fon
-family: -apple-system, "PingFang SC", "Hiragino Sans GB", Arial, &q
ot;Microsoft YaHei", "Helvetica Neue", sans-serif; font-size: medium; white-
pace: normal; background-color: rgb(255, 255, 255);">app在启动时会请求一些配置信息，其中包括ip的路由表，将这份路由表存储到本地，至于是sp还是对象持久化抑或是其他方式，可根据实际况自行选择。</p><p style="box-sizing: border-box; outline: none; line-height: 1.7em; color: rb(51, 51, 51); font-family: -apple-system, "PingFang SC", "Hiragino Sans G
", Arial, "Microsoft YaHei", "Helvetica Neue", sans-serif; font-siz
: medium; white-space: normal; background-color: rgb(255, 255, 255);">因为项目网络层由Retrfit+Rxjava+Okhttp实现，Retrofit运行时无法改变baseUrl,即使可以通过反射的方式来改变baseUrl也无法对已经生成的service对象起作用，而且我的项目中所有service对象都通过Dagger2注入，所最终使用了这样一种方式。</p><p style="box-sizing: border-box; outline: none; line-height: 1.
em; color: rgb(51, 51, 51); font-family: -apple-system, "PingFang SC", "Hirag
no Sans GB", Arial, "Microsoft YaHei", "Helvetica Neue", sans-s
rif; font-size: medium; white-space: normal; background-color: rgb(255, 255, 255);">url交给Ur
Manager来管理</p><pre class="brush:java;toolbar:false">public class UrlMana
er {
   public static final String[] HOST_SITE
 {"https://xxx/api/",xxx};
   public static final String[] HOST_WEB
 {"https://xxx/",xxx};
   public static final String HOST_SITE_DEBUG&
bsp;= "https://xxx/api/";
   public static final String HOST_WEB_DEBUG&
bsp;= "https://xxx/";
   public static List<String> list; &n
sp;
   public static Random random = new&
bsp;Random();
   public static String getHostSite() { &nb
p;
      if(BuildConfig.IS_DEBUG) { &
bsp;
         return processUrl(HO
T_SITE_DEBUG);
      } else {
        String host = ge
DynamicHost();
        if (!TextUtils.isEmpty(host))&
bsp;return host;
        return HOST_SITE[random.ne
tInt(HOST_SITE.length)];
      }
  }

  @Nullable
  private static String getDynamicHost() {
      int index = (int) SPUtils.
et(NeutronApplication.getContext(), Constants.URL, 0);
      if (list != null &&
mp; list.size() > 0 && index < list
size())
         return list.get(

ndex);
          return null;
  }
  public static String getHostWeb() {
     if (BuildConfig.IS_DEBUG) { &
bsp;
         return processUrl(HO
T_WEB_DEBUG);
     } else {
         String host =&
bsp;getDynamicHost();
         if (!TextUtils.isEmpty(
ost)) return host;
         return HOST_WEB[ra
dom.nextInt(HOST_WEB.length)];
     }
  }
   public static void setHosts(List<String> l
st) {
        UrlManager.list = list;
        RxHelper.setCounterAttempts(list.si
e());
  }
  public static void updateUrlIndex(int i) {
       if (list != null &
amp;& i >= list.size())
        i = 0;
      SPUtils.put(NeutronApplication.getContext(),Cons
ants.URL, i);
  }
   public static void updateUrlIndex() { &
bsp;
      int o = (int) SPUtils.get
NeutronApplication.getContext(), Constants.URL, 0);
       updateUrlIndex(o + 1);
 }
}</pre> <span style="color: rgb(51, 51, 51); font-family: -apple-system, "PingFang
C", "Hiragino Sans GB", Arial, "Microsoft YaHei", "Helveti
a Neue", sans-serif; font-size: medium; background-color: rgb(255, 255, 255);">app启
时拉取到配置后设置UrlManager中的路由表，然后每次根据索引去路由表动态拿请求地址，那路由
引由谁来控制呢？
<span style="color: rgb(51, 51, 51); font-family: -apple-system
"PingFang SC", "Hiragino Sans GB", Arial, "Microsoft YaHei&q
ot;, "Helvetica Neue", sans-serif; font-size: medium; background-color: rgb(255, 2
5, 255);">因为我将项目中的rxjava抽取了一层RxHelper,所以这件事就交给RxHelper来干了，可以覆
所有的网络请求。 <pre class="brush:java;toolbar:false">public class
RxHelper {
  private static final int COUNTER_START =&nb
p;0;
  private static int COUNTER_ATTEMPTS = 3;&
bsp;
  public static void setCounterAttempts(int counterAtt
mpts) {
        COUNTER_ATTEMPTS =&nb
p;counterAttempts;

```
  } 
  public static &lt;T&gt; rx.Observable.Transformer&lt;T,&nb
p;T&gt; handleResult() { 
         return tObservable&
bsp;-&gt; tObservable
             
   .flatMap(RxHelper::createData)
             
   .retryWhen(observable -&gt; observable.compose(zipWithF
atMap()))
             
   .observeOn(AndroidSchedulers.mainThread())
             
   .subscribeOn(Schedulers.io());
  }   
  public static &lt;T&gt; rx.Observable.Transformer&lt;T,T&gt
 handleResultWithOutRetryPolicy(){
        return tObserv
ble -&gt; tObservable.flatMap(RxHelper::createData)
             
   .observeOn(AndroidSchedulers.mainThread())
             
   .subscribeOn(Schedulers.io());
  }
  public static &lt;T&gt; Observable.Transformer&lt;T, 
ong&gt; zipWithFlatMap() { 
        return observable&n
sp;-&gt;
             
   observable.zipWith(Observable.range(COUNTER_START, COUNT
R_ATTEMPTS),
             
           (t, repe
tAttempt) -&gt; repeatAttempt)
             
           .flatMap(new
nbsp;Func1&lt;Integer, Observable&lt;Long&gt;&gt;() {  
             
             
  @Override
             
             
  public Observable&lt;Long&gt; call(Integer repeatAttempt
 {
             
             
      UrlManager.updateUrlIndex(repeatAttempt);&nbs
; 
             
             
      return Observable.timer(repeatAttempt&nb
p;* 200, TimeUnit.MILLISECONDS);
             
             
```

  }
                      });
  }
  private static <T> Observable<T> createData(final T t) {
         return Observable.create(new Observable.OnSubscribe<T>() {
           @Override
           public void call(Subscriber<? super T> subscriber) {
               try {
                     subscriber.onNext(t);
                     subscriber.onCompleted();
                     } catch (Exception e) {
                      LogUtils.logw("Rxhelper: " + e.toString());
                      subscriber.onError(e);
                      }
            }
        });
    }
}</pre><p style="box-sizing: border-box; outline: none; line-height: 1.7em; color: rgb(51, 51, 1); font-family: -apple-system, "PingFang SC", "Hiragino Sans GB", Ar al, "Microsoft YaHei", "Helvetica Neue", sans-serif; font-size: medium; white-space: normal; background-color: rgb(255, 255, 255);">这样在每次请求错误时，会递增由表索引，继续下次请求，轮询的间隔为<code style="box-sizing: border-box; outline: none; fon -family: Menlo, Monaco, Consolas, "Courier New", monospace; font-size: 0.8em; adding: 0.2em 0.4em; word-break: break-word; color: rgb(78, 89, 128); background-color: rgb 248, 248, 248); border-radius: 2px;">Observable.timer(repeatAttempt * 200, TimeUnit.MILLISE ONDS);</code>
对RxJava的retryWhen不理解的同学请移步对RxJava中.repeatWhen()和.retryWhen()操作符的思考。 /p><p style="box-sizing: border-box; outline: none; line-height: 1.7em; color: rgb(51, 51, 51); ont-family: -apple-system, "PingFang SC", "Hiragino Sans GB", Arial, quot;Microsoft YaHei", "Helvetica Neue", sans-serif; font-size: medium; whi e-space: normal; background-color: rgb(255, 255, 255);">之前也说了，retrofit不能修改baseUrl 反射的方式也不适合我的项目，至于利用builder生成新的retrofit对象的方式更不考虑了，那我是怎 实现运行时修改请求地址的呢？别忘了okhttp是可以添加拦截器的，在OkHttpIntercepter中：</p> pre class="brush:java;toolbar:false">public class OkHttpInterceptor impl ments Interceptor {
    @Override
    public Response intercept(Chain chain)&nbs ;throws IOException {

      //配置request
       Request request = chain.request();
        Request.Builder requestBuilder = request.newBuilder();
        String url = UrlManager.getHostSite();
        Uri parse = Uri.parse(url);
        String host = pase.getHost();
        HttpUrl httpUrl =&nbp;request.url().newBuilder().host(host).build();
        requestBuilder.url(httpUrl);
        Response.Builder responseBulder = chain.proceed(requestBuilder.build()).newBuilder();
        Response response =nbsp;responseBuilder.build();
        return response;
    }
}</pre><p style="box-sizing: border-box; outline: none; line-height: 1.7em; color: rgb(51, 51, 1); font-family: -apple-system, "PingFang SC", "Hiragino Sans GB", Aral, "Microsoft YaHei", "Helvetica Neue", sans-serif; font-size: medium; white-space: normal; background-color: rgb(255, 255, 255);">拦截请求的url,修改其host，这样个流程就ok了，http的各种错误码的处理也是可以在拦截器中统一处理的，至于其他健壮性的考虑此就不做过多阐述了。</p><p style="box-sizing: border-box; outline: none; line-height: 1.7em; color: rgb(51, 51, 51); font-family: -apple-system, "PingFang SC", "Hiragino Sas GB", Arial, "Microsoft YaHei", "Helvetica Neue", sans-serif; fon-size: medium; white-space: normal; background-color: rgb(255, 255, 255);">有同学问我，如想处理最后一次error通知怎么办呢？可以这样做，修改过的RxHelper如下：</p><pre class="brush java;toolbar:false">public class RxHelper {
     private static final int COUNTE_START = 0;
     private static int COUNTER_ATTEMPT = 3;
     public static void setCounterAttempts(int counterAttempts) {
        COUNTER_ATTEMPTS =&nbp;counterAttempts;
     }
    public static <T> rx.Observable.Transfomer<T, T> handleResult() {
           return Observable -> tObservable
             .flatMap(RxHelper::createData)
             .retryWhen(error -> delayedRetry(error))
             .observeOn(AndroidSchedulers.mainThread())
             .subscribeOn(Schedulers.io());
    }
    //猫腻主要在这个方法

```
    private static Observable&lt;Object&gt; delaedRetry(Observable&lt;? extends Throwable&gt; error) {
            retur error.zipWith(Observable.range(COUNTER_START, COUNTER_ATTEMPTS + 1),
              (i, repeatAttempt) -&gt; repeatAttempt)
              .flatMap(o -&gt; {
                   UrlManager.updateUrlIndex(o);
                   LogUtils.logw(&quot;repeat: &quot; + o);    
                   return o &lt; COUNTER_ATTEMPTS ? Observable.timer(o * 200, TimeUnit.MILLISECONDS)
                    : error.flatMap(Observable::error);
                    });
    }  
    public static &lt;T&gt; rx.Observable.Transfomer&lt;T, T&gt; handleResultWithOutRetryPolicy() {
            retur tObservable -&gt; tObservable.flatMap(RxHelper::createData)
                   .observeOn(AndroidSchedulers.mainThread())
                   .subscribeOn(Schedulers.io());
    }   
    private static &lt;T&gt; Observable&lt;T&gt;nbsp;createData(final T t) {   
                return Observable.crete(new Observable.OnSubscribe&lt;T&gt;() {    
                 @Override
                 public void call(Subscriber&lt;? super T&gt; subscriber) {&nbs; 
                   try {
                     subscriber.onNext(t);
                     subscriber.onCompleted();
                     } catch (Exception e) {
                                subscriber.onError(e);
                               }
```

           }
        });
    }
}</pre> 转自：https://gold.xitu.io/post/584e6d5961ff4b0058e9240a <hr/>
