Retrofit是square出品的一个比较有名的开源网络请求库。

现在最新的版本是2.0

下面就说说其使用方法。

1. 首先定义一个承载接收数据的接口。如下

其中@GET后面是请求数据的子URL，可以是组合的URL，如下demo即是组合URL，owner和repo是不固定的。Contributor是服务器和终端之间的通信的数据结构

public interface GitHub {

@GET("/repos/{owner}/{repo}/contributors")

Call<List<Contributor>> contributors(

@Path("owner") String owner,

@Path("repo") String repo);

}

public static class Contributor {

public final String login;

public final int contributions;

public Contributor(String login, int contributions) {

this.login = login;

this.contributions = contributions;

}

}

1. New一个client,由于demo返回的是json数据，所以我们可以用Gson去解析，retrofit本身是没有转换器的，但是square提供了一些常用数据的jar包

* [Gson](https://github.com/google/gson): com.squareup.retrofit2:converter-gson
* [Jackson](http://wiki.fasterxml.com/JacksonHome): com.squareup.retrofit2:converter-jackson
* [Moshi](https://github.com/square/moshi/): com.squareup.retrofit2:converter-moshi
* [Protobuf](https://developers.google.com/protocol-buffers/): com.squareup.retrofit2:converter-protobuf
* [Wire](https://github.com/square/wire): com.squareup.retrofit2:converter-wire
* [Simple XML](http://simple.sourceforge.net/): com.squareup.retrofit2:converter-simplexml
* Scalars (primitives, boxed, and String): com.squareup.retrofit2:converter-scalars

我们可以采用addConverterFactory把转换器加进去，如果数据格式并不在上面的常用数据格式里面，我们也可以通过实现[Converter.Factory](https://github.com/square/retrofit/blob/master/retrofit/src/main/java/retrofit/Converter.java)接口来创建一个自定义的converter 。

// Create a very simple REST adapter which points the GitHub API.

Retrofit retrofit = new Retrofit.Builder()

.baseUrl(API\_URL)

.addConverterFactory(GsonConverterFactory.create())

.build();

1. 创建接口请求实例

// Create an instance of our GitHub API interface.

GitHub github = retrofit.create(GitHub.class);

// Create a call instance for looking up Retrofit contributors.

Call<List<Contributor>> call = github.contributors("square", "retrofit");

1. 获取数据，分同步请求和异步请求

**同步请求**

|  |  |
| --- | --- |
| 1  2  3  4 | // Synchronous Call in Retrofit 2.0    List<Contributor> contributors = call.execute().body(); |

以上的代码会阻塞线程，因此你不能在安卓的主线程中调用，不然会面临NetworkOnMainThreadException。如果你想调用execute方法，请在后台线程执行。

**异步请求**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | // Synchronous Call in Retrofit 2.0    call.enqueue(List<Contributor> () {      @Override      public void onResponse(Response List<Contributor> response) {          // Get result Repo from response.body()      }        @Override      public void onFailure(Throwable t) {        }  }); |

以上代码发起了一个在后台线程的请求并从response 的response.body()方法中获取一个结果对象。注意这里的onResponse和onFailure方法是在主线程中调用的。