

转载请注明链接:

讲解OkHttp封装中的要点，最终封装成SDK，作为底层sdk供后续使用。

如果有帮助的话，请点个赞！万分感谢！

OkHttp源码详解

版本：2018/9/3-1(23:20)

- [OkHttp源码详解](#)
 - [为什么要封装OkHttp](#)
 - [模块划分](#)
 - [Request](#)
 - [RequestParams](#)
 - [CommonRequest](#)
 - [Callback](#)
 - [CommonCallBack](#)
 - [Listener](#)
 - [OkHttpClient](#)
 - [知识扩展](#)

为什么要封装OkHttp

1. 为什么要封装OkHttp?

1. 代码冗余，不可复用：所有用到网络请求的地方，都要去额外编写Request、OkHttpClient、Call等代码，会导致代码冗余。
2. okhttp API在版本更新后，可能会改变，不利于维护：网络请求直接使用OkHttpApi会导致一旦第三方开源库的API发生改变，就需要大面积修改代码。

模块划分

2、OkHttp封装的思路

1. OkHttp的使用，主要是三部分：根据请求参数去构建Request、配置OkHttpClient并且去发送get和poest请求、以及获取到数据后通过Callback回调给上层。
2. 此外需要考虑https的支持、重定向的支持。

3、OkHttp封装的模块(6)

1. exception: `OkHttpException`,如果出现了网络异常,解析异常,通过`onFailure()`将该对象交给上层。
2. Callback: 监听器,用于将结果返回给数据。使用`CommonJsonCallback`对结果有Json时,解析后在通过`Listener`交给上层。
3. cookie: 处理cookie
4. https: `HttpsUtils`, 返回`SslSocketFactory`
5. Request: `CommonRequest`、`RequestParams`, 去构造需要的Request
6. `CommonOkHttpClient`: 初始化`OkHttpClient`的必要参数, 去设置超时, 支持https, 支持重定向。然后通过该类进行网络请求。

4、OkHttp封装需要考虑哪些事情?

1. https的支持
2. 异常情况处理
3. 结果处理: 将json转换为对应实体
4. 结果转发: 将结果转到UI线程

Request

1、Request封装分两部分

1. `RequestParams`: 封装所有的请求参数到`HashMap`中
2. `CommonRequest`: 接收请求参数, 为我们生成Request对象

RequestParams

1、RequestParams包含了哪些内容?

1. 用`ConcurrentHashMap`去存储url和file的参数。
2. 提供put方法, 去存放两种类型的参数

CommonRequest

1、CommonRequest

1. 根据url和params构造出Get或者Post的Request对象

Callback

CommonCallBack

1、要做哪些工作？

1. 结果处理
2. 异常处理
3. 结果转发

Listener

OkHttpClient

1、CommonOkHttpClient

1. static代码块中，去设置固定的超时时间参数(连接、读取、写入)。这样所有的对象都进行了配置

2、如何支持Https请求

1. 验证身份

```
// https,支持官方和自己的https请求
builder.hostnameVerifier(new HostnameVerifier() {
    @Override
    public boolean verify(String hostname, SSLSession session) {
        // 这里应该进行验证，这里就暂时返回true
        return true;
    }
});
```

3、http请求，如果页面销毁的，结果返回会浪费资源，应该在页面销毁时，去取消请求。

知识扩展

1、如何遍历HashMap？

map.entrySet()能通过迭代器进行遍历

2、字符拼接应该使用StringBuilder

3、SSLSocketFactory是什么？

1. 使用Https时需要生成SSL的socket
2. 普通请求就是普通SocketFactory返回的socket

4、okhttp有callback为什么额外封装？

> 1. okhttp的api可能会变化，需要进一遍封装

> 1. 不便于扩展(比如下载进度监听)

5、windows charles Http请求调试