



河北师范大学软件学院
Software College of Hebei Normal University

OkHttp实战



Android教研室



OkHttp简介



OkHttp请求处理流程



环境配置



核心API



主要功能





OkHttp简介

- OkHttp是square公司推出的一款Android和Java网络请求库，是Android目前最流行的网络库之一，当前版本是OkHttp3。
- OkHttp3支持Android2.3及以上版本。
- 支持Java JDK 1.7以上版本。



OkHttp的优势

- 支持对同一主机的所有请求共享一个套接字
- 通过连接池来减少请求延迟
- 透明的GZIP压缩了下载文件的大小
- 缓存响应数据来减少重复的网络请求
- 减少了对数据流量的消耗



OkHttp的功能

- 请求文本数据
- 文件的上传
- 文件的下载
- 加载图片（自动压缩图片大小）



OkHttp简介



OkHttp请求处理流程



环境配置



核心API



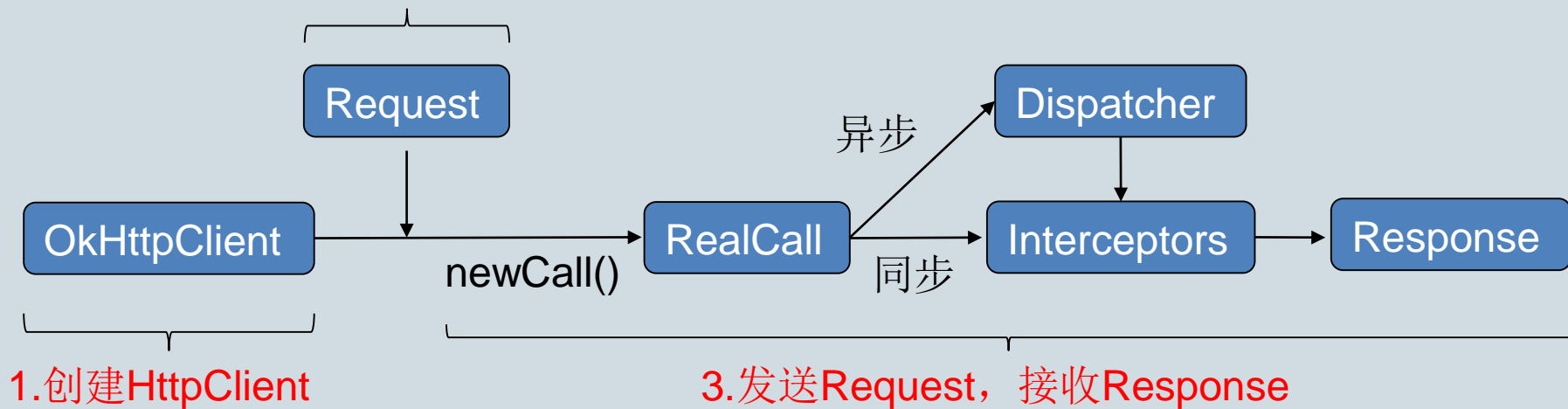
主要功能





OkHttp请求处理流程

2. 创建Request请求





OkHttp简介



OkHttp请求处理流程



环境配置



核心API



主要功能





配置环境

➤ Gradle中添加依赖

```
dependencies {  
    .....  
    //OkHttp框架  
    implementation 'com.squareup.okhttp3:okhttp:3.10.0'  
}
```



配置环境

- Manifest.xml添加访问网络的权限

```
<uses-permission  
    android:name="android.permission.INTERNET"/>
```



OkHttp简介



OkHttp请求处理流程



环境配置



核心API



主要功能





OkHttp的核心API

➤ OkHttpClient类

- OkHttpClient对象的创建方式有两种。第一种是使用构造方法创建对象：

```
OkHttpClient okHttpClient = new OkHttpClient();
```



OkHttp的核心API

➤ OkHttpClient类

- 如果要自定义一些属性需要使用Builder模式创建对象：

```
OkHttpClient okHttpClient = new OkHttpClient.Builder()  
    .cache() // 缓存目录  
    .connectTimeout() // 连接超时  
    .cookieJar() // cookie的读取，保存，需要自己实现  
    .build();
```



OkHttp的核心API

➤ Request类

- 使用OkHttp创建Request请求包含以下几个部分：

设置请求地址：HttpUrl

设置请求方式：默认为GET请求

构造请求头：Headers

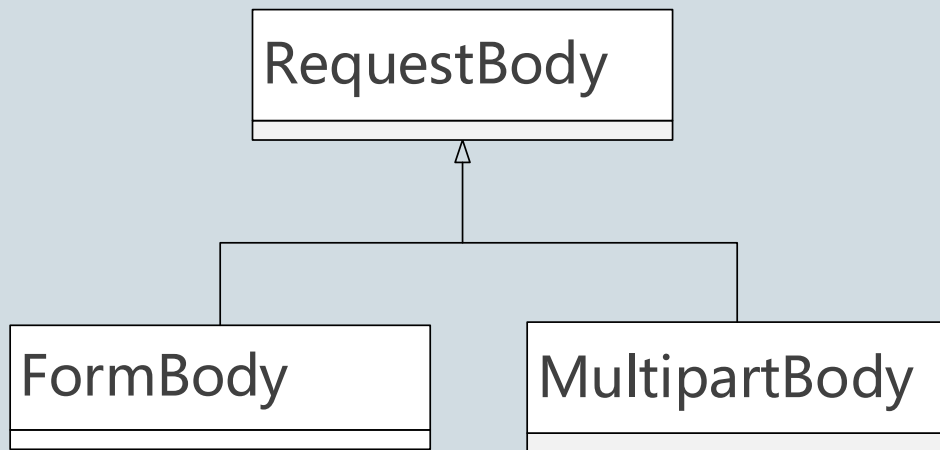
构造请求体：RequestBody



OkHttp的核心API

➤ RequestBody类

- 上传数据的核心类，请求体有文件、JSON字符串等多种形式，通常用的是FormBody和MultipartBody





OkHttp的核心API

➤ Request对象创建方法

- 利用RequestBuilder创建Request请求对象

```
Request.Builder builder = new Request.Builder();  
builder.url(url);  
Request request = builder.build();
```




OkHttp的核心API

➤ Call接口

- 在发起请求时，整个框架主要通过Call来封装每一次的请求。
- RealCall是Call接口的实现类。

```
Call call = okHttpClient.newCall(request);
```



OkHttp的核心API

➤ Response类

- 请求返回的响应码(成功的200或没有找到内容的404)，头信息，和可选的实体。



OkHttp的核心API

➤ 基本操作步骤

- 创建OKHttpClient对象
- 创建Request请求对象
- 创建Call对象
- 提交请求并返回响应



- OkHttp简介
- OkHttp请求处理流程
- 环境配置
- 核心API
- 主要功能





OkHttp实现get请求

//1. 构建一个*OkHttpClient*

```
OkHttpClient okHttpClient = new OkHttpClient();
```

//2. 构建一个*Request*

```
Request request = new
```

```
    Request.Builder().url("http://baidu.com").build();
```

//3. 获得*Call*对象

```
final Call call = okHttpClient.newCall(request);
```



OkHttp实现get请求

➤ 同步方式实现

```
new Thread(new Runnable() {  
    @Override  
    public void run() {  
        try {  
            Response response = call.execute(); //1) 同步请求  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}).start();
```



OkHttp实现get请求

➤ 异步方式实现

```
call.enqueue(new Callback() {//2) 异步请求  
    @Override  
    public void onFailure(Call call, IOException e) {  
        //发生错误时执行的回调  
    }  
    @Override  
    public void onResponse(Call call, Response r) throws IOException {  
        //正确执行, 获取返回数据的回调  
    }  
});
```



OkHttp实现post请求

➤ `RequestBody.create(MediaType contentType, String content)`

- `contentType`表示请求的编码方式
- 参考网址：

http://www.w3school.com.cn/media/media_mime_ref.asp



OkHttp实现post请求

➤ 提交JSON格式数据

```
RequestBody requestBody = RequestBody.create(  
    MediaType.parse("text/plain;charset=utf-8"),  
    "{ \"userName\": \"zhangsan\", \"password\": \"123\" }");  
  
Request request = new Request.Builder()  
    .post(requestBody)  
    .url(BASE_URL + "postStringAction").build();
```



OkHttp实现post请求

➤ 提交键值格式数据

```
FormBody.Builder builder = new FormBody.Builder();  
builder.add("userName", "zhangsan");  
builder.add("password", "123");  
FormBody body = builder.build();
```

```
Request request = new Request.Builder().post(body)  
    .url(BASE_URL + "testAction").build();
```



OkHttp实现文件上传

```
File file = new File(Environment.  
    getExternalStorageDirectory(), "test.jpg");  
RequestBody requestBody = RequestBody.create(  
    MediaType.parse("application/octet-stream"), file);  
  
Request.Builder builder = new Request.Builder();  
Request request = builder  
    .post(requestBody)  
    .url(BASE_URL + "uploadFileAction")  
    .build();
```



OkHttp实现文件上传

- Manifest.xml添加访问SD卡的权限

```
<uses-permission android:name=  
    "android.permission.WRITE_EXTERNAL_STORAGE"/>
```



OkHttp实现文件下载

//1. 构建一个OkHttpClient

```
OkHttpClient okHttpClient = new OkHttpClient();
```

//2. 构建一个Request

```
Request request = new Request.Builder()  
    .url(BASE_URL + "files/test.txt")  
    .build();
```

//3. 获得Call对象

```
final Call call = okHttpClient.newCall(request);
```



OkHttp实现文件下载

@Override

```
public void onResponse(Call call, Response r) throws IOException {  
    InputStream is = response.body().byteStream();  
    File file = new File(Environment.getExternalStorageDirectory(),  
                           "test.txt");  
    FileOutputStream fos = new FileOutputStream(file);  
    int len = 0;  
    byte[] buf = new byte[128];  
    while ((len = is.read(buf)) != -1) {  
        fos.write(buf, 0, len);  
    }  
    .....  
}
```



Thank You!

