



电子信息工程学院

移动终端开发技术

# 网络编程的最佳实践

讲 师：陈媛媛

# 最佳实践



因为每一个应用程序很有可能会在很多地方都使用到网络功能，而发送 HTTP 请求的代码基本都是相同的，我们每次都去编写一遍发送 HTTP 请求的代码吗？

解决方法：通常情况下我们都应该将这些通用的网络操作提取到一个公共的类里，并提供一个静态方法，当想要发起网络请求的时候只需简单地调用一下这个方法即可。

# 最佳实践



网络请求通常都是属于耗时操作，而 `sendHttpRequest()` 方法的需要开启线程，否则有可能导致在调用 `sendHttpRequest()` 方法的时候使得主线程被阻塞住。但是开启了一个线程来发起 HTTP 请求，那么服务器响应的数据是无法进行返回的。

解决方法：需要使用 Java 的回调机制。

# 最佳实践

首先需要定义一个接口，比如将它命名成 `HttpCallbackListener`，代码如下所示：

```
public interface HttpCallbackListener {  
    void onFinish(String response);  
    void onError(Exception e);}
```

接着修改 `HttpUtil` 中的代码，如下所示：

```
public class HttpUtil {  
    public static void sendHttpRequest(final String address, final  
    HttpCallbackListener listener) {...}}
```

# 最佳实践

最后在调用它的时候还需要将 `HttpCallbackListener` 的实例传入，如下所示：

```
HttpUtil.sendHttpRequest(address, new HttpCallbackListener() {  
    @Override  
    public void onFinish(String response) {  
        // 在这里根据返回内容执行具体的逻辑 }  
    @Override  
    public void onError(Exception e) {  
        // 在这里对异常情况进行处理  
    }  
});
```



1

使用HTTP协议访问网络

---

2

解析XML格式数据

---

3

解析JSON格式数据

---