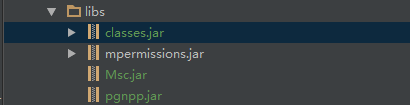
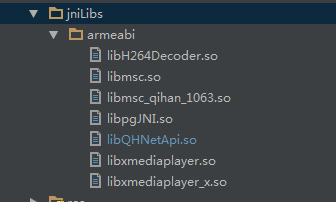
**QLinkDemo Android接口文档**

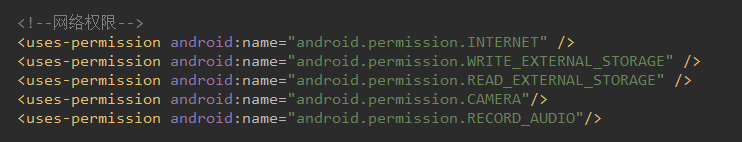
1. 搭建开发环境（Android Studio环境）
   1. 在Android Studio 建立你的工程
   2. 把demo工程libs目录的jar包复制到你的工程，并引入以下Jar库，classes.jar是封装库，源码在mylibrary工程中，msc和pgnpp是视频库，mpermissions 是第3方动态权限申请库。



* 1. 在项目src/main 目录下建立jniLibs文件夹，把demo 工程中的\*.so后缀文件复制到jniLibs文件夹下。



* 1. 权限配置，在AndroidManifest.xml 添加以下权限。



1. 开始使用
   1. 初始化网络库，非阻塞，程序启动时调用

**public native int initLib(int app\_id, String domain, int port, String version);**

app\_id 保存在清单文件中，用来区分不同的版本渠道

domain 服务器IP

Port 服务器端口

Version 当前版本

Returen 非0即错误码（错误码查询在NetInfo文件中）

* 1. 反初始化网络库，阻塞型，程序结束时调用。

Returen 非0即错误码

**public native int cleanupLib();**

1. 如何登录

**public native int onLogin(LoginAccount loginAccount);**

参数说明：

LoginAccount class

accountType：帐号类型qlink\_id/tel/mail

account：帐号

password:密码

回调参数：NetInfo.QHC\_CMD\_LOGIN\_RSP

返回参数：

返回0 成功 其他为错误码

1. 如何处理回调

public void showRecv(int cmd, int result, Object obj, long seq)

public void showVideo(int handle, byte[] data, int len, int width, int height, int sec, int type)

在NetApi中，有2个回调方法，showRecv处理常用的方法回调，showVideo处理视频回调。

**void showRecv(int cmd, int result, Object obj, long seq);**

cmd: 回调的命令字

result: 错误码 0-成功 其他-错误

obj: 由JNI层请求网络返回的数据

seq：请求码，与每个请求的seq值对应

**void showVideo(int handle, byte[] data, int len, width, int height, int sec, int type);**

handle:打开视频时的会话

data:音频数据（PCM数据）或者视频数据（RGB565图片数据）

len:数据长度

width：视频宽度

height:视频高度

type:数据类型 1 I帧 2 P帧 3音频