NETCA KeyX SDK接口说明

# 简介

在Android平台上NETCA KeyX接口以aar文件的形式提供相应的PKI功能。

# 术语

NetcaKeyX：网证通安全客户端Android版。

第三方应用：指使用该SDK的应用APP。

# 使用步骤

1. 导入SDK文件
   1. 通过模块导入 netca\_sdk 目录
   2. app项目依赖增加 implementation project(':netca\_sdk')
   3. setting.gradle 增加 ':netca\_sdk'

注意：本SDK支持Android API 17及以上。

1. 增加外部依赖
   1. implementation 'com.alibaba:fastjson:1.1.56.android'
   2. implementation 'com.squareup.okhttp3:okhttp:3.5.0'
2. 增加自定义权限

<uses-permission android:name="net.netca.pki.crypto.service.permission.aidl" />

1. 初始化

在onResume中加入

GlobalSetting.getInstance().init(this, callback);

1. 签名/登录
2. try {  
    //方法调用必须在\*线程\*中  
    //多证书时，网证通客户端需要悬浮窗权限  
    Cert mCert = Cert.*selectSingleCert*(Cert.*CERT\_TYPE\_SIGNATURE*);  
    //随机数由服务器参数，用户对该随机数进行签名，后台验证签名，拿出签名证书，找到证书绑定的用户，使用该用户登录。  
    String clearData = "random data";  
    //密码。  
    String psw = "123456";  
    byte[] clear = clearData.getBytes("utf-8");  
    if (mCert == null) {  
    Log.*i*("cert", "未找到签名相关证书");  
    return;  
    } else {  
    SignedDataSign sign = new SignedDataSign();  
    boolean isDetach = false;  
    String signedDataMethod = Signature.*SHA256WITHRSA*;  
    if (mCert.isSm2()) {  
    signedDataMethod = Signature.*SM3WITHSM2*;  
    }  
    signedData = sign.signedDataSign(mCert, clear, signedDataMethod, isDetach, SignedDataSign.*INCLUDE\_CERT\_OPTION\_SELF*, psw, null);  
    //该签名值需传给后台进行验证  
    String data = Base64.*encodeToString*(signedData, Base64.*NO\_WRAP*);  
    }  
   }catch (Exception e){  
    e.printStackTrace();  
   }

# 权限

网证通安全客户端 需要悬浮窗权限，关联启动权限，打开 网证通安全客户端 并授予该权限。

第三方应用需要 网络权限，用于OCSP CRL 验证。