1 LibJeid サンプルアプリ

Jeid ライブラリを使ったサンプルアプリです。
ディレクトリ構成

- app アプリ本体
- libjeid-full/libjeid-full-release.aar Jeid ライブラリ

1.1 API レベル

- android.nfc.tech.IsoDep
  - Jeid ライブラリの要求レベル
  - API レベル 10(Android 2.3) から
- android.provider.Settings.ACTION_NFC_SETTINGS
  - API レベル 16(Android 4.1) から
  - NFC の設定画面を開く為に必要
- Webview#evaluateJavascript()
  - API レベル 19(Android 4.4) から
  - 免許証のレンダリングに利用

このサンプルアプリは minSdkVersion:19 を利用しますが、読み取り機能だけを実装する場合は API レベル 10 でも開発可能です。

2 ビルド手順

1. Java 開発環境のインストール

Debian/Ubuntu:

```
$ sudo apt install openjdk-8-jre
```

RHEL/CentOS:

```
$ sudo yum install java-1.8.0-openjdk-devel
```

2. Android Studio(JDK 付属) または JDK+Android SDK をインストールします。

以下は Android SDK のインストール

```
$ wget https://dl.google.com/android/repository/sdk-tools-linux-4333796.zip
# mkdir -p /opt/android/sdk/
# unzip sdk-tools-linux-4333796.zip -d /opt/android/sdk/
```

ライセンスに同意

```
$ yes | /opt/android/sdk/tools/bin/sdkmanager --licenses
```

3. local.properties に SDK のパスを設定

```
$ echo "sdk.dir=/opt/android/sdk" > local.properties
```

4. ビルド

```
$ ./gradlew build
```

- デバッグビルド ./app/build/outputs/apk/debug/app-debug.apk
- リリースビルド (未署名) ./app/build/outputs/apk/release/app-release-unsigned.apk

2.1 サンプルコード

2.1.1 券面事項入力補助 AP 券面事項取得

```
try {
    String pin = "XXXX";
    JeidReader reader = new JeidReader(nfcTag);
    CardInputHelperAP ap = reader.selectCardInputHelperAP();
    ap.verifyPin(pin);
    String mynumber = ap.getMyNumber();
    System.out.println("マイナンバー: " + mynumber);
    CardInputHelperEntries entries = ap.getEntries();
    System.out.println("氏名        : " + entries.getName());
    System.out.println("住所        : " + entries.getAddr());
    System.out.println("生年月日    : " + entries.getBirth());
    System.out.println("性別        : " + entries.getSexString());
}catch(InvalidPinException e){
    if (e.isBlocked()) {
        System.out.println("PIN がブロックされています。");
    } else {
        System.out.println("PIN が間違っています。残り回数: " + e.getPinCounter());
    }
}catch(IOException e){
    System.out.println("読み取りエラー");
}
```

2.1.2 公的個人認証 AP

```
try {
    // ユーザー認証用暗証番号
    String authPin = "XXXX";
    // デジタル署名用パスワード
    String signPin = "XXXXXXXX";

    JeidReader reader = new JeidReader(nfcTag);
    JPKIAP jpki = reader.selectJPKIAP();

    // ユーザー認証用証明書を取得
    X509Certificate authCert = jpki.getAuthCert();

    // ユーザー認証用鍵で署名
    JPKISignature authSignature = jpki.getAuthSignature("SHA1withRSA");
    authSignature.update("hello".getBytes());
    byte[] signed = authSignature.sign(authPin);

    // デジタル署名用証明書を取得
    jpki.verifyPassword(signPin);
    X509Certificate signCert = jpki.getSignCert();

    // デジタル署名用鍵で署名
    JPKISignature signSignature = jpki.getSignSignature("SHA1withRSA");
    signSignature.update("hello".getBytes());
    signed = signSignature.sign(signPin);

}catch(InvalidPinException e){
    if (e.isBlocked()) {
        System.out.println("PIN がブロックされています。");
    } else {
        System.out.println("PIN が間違っています。残り回数: " + e.getPinCounter());
    }
}catch(IOException e){
    System.out.println("読み取りエラー");
}
```

3 逆引きリファレンス

- 券面事項を取得する
  - app/src/main/java/jp/co/osstech/jeidreader/CardInfoActivity.java
  - app/src/main/java/jp/co/osstech/jeidreader/CardInfoTask.java
- JPKI 証明書を取得・表示する
  - app/src/main/java/jp/co/osstech/jeidreader/ShowCertActivity.java
  - app/src/main/java/jp/co/osstech/jeidreader/ShowCertTask.java
- JPKI で署名する
  - app/src/main/java/jp/co/osstech/jeidreader/SignActivity.java
  - app/src/main/java/jp/co/osstech/jeidreader/SignTask.java
- PIN ステータスを取得する
  - app/src/main/java/jp/co/osstech/jeidreader/PinStatusActivity.java
  - app/src/main/java/jp/co/osstech/jeidreader/PinStatusTask.java

4 JPKI 署名検証

「JPKI 認証用署名」および「JPKI 署名用署名」を実行すると下記 4 つのファイルを端末内に保存します。

- input.txt: 署名対象を Base64 エンコードしたもの
- digest.txt: ハッシュ値を Base64 エンコードしたもの
- cert.txt: JPKI 認証用証明書 (DER) を Base64 エンコードしたもの
- signed.txt: 署名値を Base64 エンコードしたもの

これらのファイルは Android 端末を PC につないで以下のフォルダから MTP 転送できます。

```
/Android/data/jp.co.osstech.jeidreader/files/
```

これらのファイルを用いて OpenSSL コマンドで簡易的な署名検証ができます。

```
# 署名対象を Base64 デコード
$ base64 -d input.txt > input.dat
```

```
# 証明書から公開鍵を取り出し
$ base64 -d cert.txt | openssl x509 -inform der -noout -pubkey > pub.pem

# 署名値を Base64 デコード
$ base64 -d signed.txt > signed.dat

# 公開鍵による署名検証
$ openssl dgst -sha1 -verify pub.pem -signature signed.dat input.dat
Verified OK
```