

系统设置与 FaceId 交互接口说明

一. 设置开关以及查询接口

1.1 集成 vis-ext-client.aar

1.2 初始化

`VisClient.instance().init(Application context, VisClientStateListener listener)`

该方法须在 application 的 `onCreate()` 中调用，`VisClientStateListener` 是监听和 AIDL 服务端连接/绑定相关状态的，它的回调方法如下，应用可以根据自己需要使用：

```
/**
 * FaceId AIDL 服务端建立连接
 */
public void onServiceConnected() {
}

/**
 * FaceId AIDL 服务端断开连接
 */
public void onServiceDisconnected() {
}

/**
 * FaceId AIDL 服务端绑定失败
 */
public void onBindServiceFailure() {
}

/**
 * FaceId AIDL 服务端绑定成功
 */
public void onBindServiceSuccess() {
}

/**
 * ClientService 注册到 FaceId AIDL 服务端注册失败
 */
public void onRegisterClientServiceFailure() {
}
```

```

/**
 * FaceId AIDL 服务端解绑
 */
public void onUnbindService() {
}

/**
 * 发送请求给 FaceId AIDL 服务端成功
 */
public void onSendSuccess(Bundle intent, String result) {
}

/**
 * 发送请求给 FaceId AIDL 服务端失败
 */
public void onSendFailure(Bundle intent) {
}

```

1.3 开启/关闭 FaceId

FaceIdExtAbility.setAbility(boolean enable)

true: 开启; false: 关闭

1.4 查询接口

FaceIdExtAbility.query(), 使用该接口向 FaceId 发送查询请求

1.5 监听结果

应用需要创建需要创建 FaceIdSwitchReceiver, 继承自 AbsVisSwitchReceiver, 用来接收开关和查询结果。

```

public class FaceIdSwitchReceiver extends AbsVisSwitchReceiver {

    /**
     * 接收 FaceId 开关消息结果
     * @param type 消息类型: 设置开关结果, VisExtConfig.ABILITY_SWITCH = 1 ;
     * 查询开关结果    VisExtConfig.ABILITY_QUERY = 2
     * @param code 状态码: 成功 VisExtConfig.SUCCESS = 0; 失败 VisExtConfig.FAIL = -1
     * @param map 消息类型: ABILITY_SWITCH 时, map 内容为设置后的开关状态。
     */
}

```

```

* 消息类型为: ABILITY_QUERY,      map 内容为查询到的开关状态,
*/

@Override
protected void onReceiveFaceId(int type, int code,
HashMap<String, Boolean> map) {
    switch (type) {
        case VisExtConfig.ABILITY_SWITCH:
            //设置开关的结果
            break;
        case VisExtConfig.ABILITY_QUERY:
            //查询开关的结果
            break;
        default:
            break;
    }
}

```

1.6 注册和反注册,

FaceIdReceiver.register(AbsVisExtReceiver receiver) 注册监听

FaceIdReceiver.unregister() 反注册

二. FaceId 介绍说明

介绍说明弹窗的实现在 FaceId 应用这边, 设置需要通过 intent 来启动 FaceId 应用这边的弹窗。启动方式参考如下:

```

private void startDialogActivity() {
    Intent intent = new Intent("com.baidu.iov.vision.show.dialog");
    intent.putExtra("type", "type_introduction_faceid");
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}

```

三. FaceId 隐私协议详情

隐私协议详情页面, 实现也在 faceID 应用这边, 可以通过 intent 来启动。

代码参考如下:

```

private void startPrivacyIntroductionActivity() {
    Intent intent = new Intent("com.baidu.iov.vision.privacy");
    intent.putExtra("vision_type", 1);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}

```