|  |  |
| --- | --- |
|  | **iOnePass**  **for Android**  **개발 가이드** |

**IDEATEC Co, Ltd.**

목 차

[1 개 요 3](#_Toc41246205)

[1.1 소 개 3](#_Toc41246206)

[1.2 주의사항 3](#_Toc41246207)

[1.3 개발지원 3](#_Toc41246208)

[2 구 성 3](#_Toc41246209)

[2.1 시스템구성 및 흐름도 3](#_Toc41246210)

[3 프로젝트 환경설정 3](#_Toc41246211)

[3.1 라이브러리 구성 3](#_Toc41246212)

[3.2 프로젝트 설정 4](#_Toc41246213)

[3.2.1 Android Studio에서 프로젝트 설정. 4](#_Toc41246214)

[3.2.2 보안 네트워크 통신 정책 변경 4](#_Toc41246215)

[4 iOnePassLib 개발 가이드 4](#_Toc41246216)

[4.1 호출 흐름도 4](#_Toc41246217)

[4.1.1 객체생성 5](#_Toc41246218)

[4.1.2 등록 단계 5](#_Toc41246219)

[4.1.2.1 요청 방법 5](#_Toc41246220)

[4.1.2.2 onActivityResult 처리 5](#_Toc41246221)

[4.1.3 인증 단계 6](#_Toc41246222)

[4.1.3.1 요청 방법 6](#_Toc41246223)

[4.1.3.2 onActivityResult 처리 6](#_Toc41246224)

[4.1.4 해제 단계 7](#_Toc41246225)

[4.1.4.1 요청 방법 7](#_Toc41246226)

# **1 개 요**

## **1.1 소 개**

|  |
| --- |
| iOnePassLib은 Fido 생체인증 및 Pin, Pattern을 2차 인증수단으로 제공하는 종합 솔루션으로, 고객사의 어플리케이션에 쉽게 적용하여 개발하도록 가이드를 제공합니다. |

## **1.2 주의사항**

|  |
| --- |
| - Android SDK Version 최소 설정 21이상 |

## **1.3 개발지원**

|  |
| --- |
| 이데아텍 보안사업부 연락처  E-MAIL : hanit@ideatec.co.kr  TEL : 02-4060-3868 |

# **2 구 성**

## **2.1 시스템구성 및 흐름도**

|  |
| --- |
|  |

※ 고객사 앱에서 필요시 호출하면 인증 후 결과를 고객사 앱으로 (성공, 실패)여부를 전달합니다.

# **3 프로젝트 환경설정**

## **3.1 라이브러리 구성**

|  |
| --- |
| iOnePass\_AOS\_1.X.XX.yyyyMMddii  ├doc : 각종 가이드 문서가 있습니다.(본 문서 포함)  ┖library : Android용 라이브러리 패키지입니다. |

## **3.2 프로젝트 설정**

### 3.2.1 Android Studio에서 프로젝트 설정.

|  |
| --- |
| **iOnePassLib for Android SDK 설치하기**  1. 제공받은 iOnePassLib-1.x.x.zip 압축 파일에서 Android용 iOnePassLib SDK 압축 파일을 풉니다.  2. iOnePassLib.aar 파일을 프로젝트 root\app\libs 폴더에 복사를 합니다.  3. Android Studio를 열고 작업할 프로젝트를 불러 옵니다. build.gradle (Module:app) 파일에 아래 내용을 추가합니다.  repositories **{** flatDir **{** dirs 'libs' **} }**  dependencies **{**  …  implementation 'kr.co.ideatec.ionepasslib:iOnePassLib:1.0@aar'  … |

### 3.2.2 보안 네트워크 통신 정책 변경

일부 HTTP통신을 사용할 필요가 있는 경우에는 AndroidManifast.xml에 아래 내용을 추가 합니다.

|  |
| --- |
| <uses-library android:name="org.apache.http.legacy" android:required="false"/> |

더 자세한 “iOnePassLib 환경설정” 내용은을 참고하세요.

# **4 iOnePassLib 개발 가이드**

## **4.1 호출 흐름도**

|  |
| --- |
|  |

※ 2차인증 모듈 호출은 환경설정 뒤 결과상태를 받을 Listener를 정의 및 설정, 결과를 수신하여 이후 작업을 하는 순서로 진행됩니다. 아래 단계별 예제 소스와 추가 설명이 기입되어있습니다.

### 4.1.1 객체생성

|  |
| --- |
| import kr.co.ideatec.ionepasslib.IOnePassLib; import kr.co.ideatec.ionepasslib.listener.iOnePassLibNetworkListener;   public class RegisterActivity extends AppCompatActivity { …   IOnePassLib mIOPLib;  private static final String *TAG* = RegisterActivity.class.getSimpleName();   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_register*);  this.initializeView();  this.initEvent();   }   public void initializeView() {//객체생성  …  mIOPLib = IOnePassLib.getInstance();  }  … |

※ 상기코드는 샘플 앱에 구현한 것 중 일부만을 표시 한 것입니다.

### 4.1.2 등록 단계

### 4.1.2.1 요청 방법

|  |
| --- |
| …  mIOPLib.reqRegistration(RegisterActivity.this, username,new iOnePassLibNetworkListener() { // 등록작업시에 .reqResigstration을 호출한다.  @Override  public void didError(String errMessage) {  Toast.*makeText*(RegisterActivity.this.getApplicationContext(),"등록실패 : " + errMessage,Toast.*LENGTH\_SHORT*);  }   @Override  public void didSuccess() {  Toast.*makeText*(RegisterActivity.this.getApplicationContext(),"등록성공",Toast.*LENGTH\_SHORT*);  }  });  }  … |

※ 등록 단계시 필요한 메소드를 호출 합니다. mIOPLib.reqResistration(); 자세한 내역은 별첨으로 제공된 API Reference를 참고해 주세요.

### 4.1.2.2 onActivityResult 처리

|  |
| --- |
| …  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  super.onActivityResult(requestCode, resultCode, data);  Toast.*makeText*(this, "REQUEST CODE : " + requestCode + " / RESULT CODE : " + resultCode, Toast.*LENGTH\_SHORT*).show();   mIOPLib.sendUAFRegResponseData(this, requestCode, resultCode, data,new iOnePassLibNetworkListener() {   @Override  public void didSuccess() {   Log.*d*(*TAG*, "IOP\_FIDO\_RP\_CLIENT.handleSuccess : ");  }   @Override  public void didError(String errorMessage) {  Log.*d*(*TAG*, "IOP\_FIDO\_RP\_CLIENT.handleError : " + errorMessage);  }  });  Toast.*makeText*(RegisterActivity.this.getApplicationContext(),"등록성공",Toast.*LENGTH\_SHORT*);  }  });  }  … |

※ 등록 단계시 필요한 메소드를 호출 합니다. mIOPLib.reqResistration(); 자세한 내역은 별첨으로 제공된 API Reference를 참고해 주세요.

### 4.1.3 인증 단계

### 4.1.3.1 요청 방법

|  |
| --- |
| …  mIOPLib.reqAuth(mContext,username,transMessage, new iOnePassLibNetworkListener() {  @Override  public void didError(String errMessage) {  Toast.*makeText*(mContext,"등록실패 : " + errMessage, *LENGTH\_SHORT*);  }   @Override  public void didSuccess() {  Toast.*makeText*(mContext,"등록성공", *LENGTH\_SHORT*);  } });  … |

※ 인증 단계시 필요한 메소드를 호출 합니다. mIOPLib.reqAuth(); 자세한 내역은 별첨으로 제공된 API Reference를 참고해 주세요.

### 4.1.3.2 onActivityResult 처리

|  |
| --- |
| …  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  super.onActivityResult(requestCode, resultCode, data);  Toast.*makeText*(this, "REQUEST CODE : " + requestCode + " / RESULT CODE : " + resultCode, *LENGTH\_SHORT*).show();   mIOPLib.sendUAFAuthResponseData(requestCode, resultCode, data, new iOnePassLibNetworkListener() {   @Override  public void didSuccess() {  Log.*i*(*TAG*,"onActivityResult >> sendUAFAuthResponseData >> Success ");  }   @Override  public void didError(String errorMessage) { // CustomLog.d(TAG, "IOP\_FIDO\_RP\_CLIENT.handleError : " + errorMessage); // CommonAlertDialog.showOkDialog(AuthActivity.this, errorMessage);  }  });    … |

### 4.1.4 해제 단계

### 4.1.4.1 요청 방법

|  |
| --- |
| …  mIOPLib.reqDeregistration(mContext, username, new iOnePassLibNetworkListener() {  @Override  public void didError(String errMessage) {  Toast.*makeText*(mContext,"등록실패 : " + errMessage,Toast.*LENGTH\_SHORT*);  }   @Override  public void didSuccess() {  Toast.*makeText*(mContext,"등록성공",Toast.*LENGTH\_SHORT*);  } });  … |

※ 해제 단계시 필요한 메소드를 호출 합니다. mIOPLib.reqAuth(); 자세한 내역은 별첨으로 제공된 API Reference를 참고해 주세요.