Table of Contents

Introduction

본 문서는 Coinstack Openkeychain에서 안드로이드 지문 인증 모듈에 대해서 기술합니다.

개요

Coinstack의 OpenKeyChain은 블럭체인 기반의 KeyChain 시스템으로 애플리케이션에서 필요한 PKI 및 인증서를 생성, 등록, 인증을 할 수 있다. 본 문서는 Android 어플리케이션에서 OpenKeyChain에 사용되는 개인키를 안전하게 보호하기 위해 Android의 지문인증 기능과 키 저장소를 사용한 지문인증 모듈에 관해 설명합니다.

기술 지원

해당 모듈을 사용해서 개발에 어려움이 있다면 언제든지 아래의 연락처를 통해 문의하길 바랍니다.

> support@blocko.io
>
> +82+031-8016-6253
>
> https://blocko.io

본 문서에 관하여

본 문서는 오픈 키체인 시스템에서 안드로이드 클라이언트의 지문인증 모듈에 관한 문서로 다음의 내용을 포함하고 있습니다.

- 인증 모듈의 동작 원리
- 인증 모듈을 사용한 인증 절차 및 샘플
- 인증 모듈을 사용하는데 주의해야 할 요소들
- 사용자화를 위한 가이드라인

기술 구조

암호화

기본 원리의 이해

> Alice와 Bob은 같은 학급의 친한 친구인데, 어느날 전화 통화를 하던 중 내기를 하기로 하였다. Alice는 전화번호부에서 "02-1234-5678" 번호가 포함되어 있는 페이지가 홀수인지 짝수인지 맞추면 이기는 게임을 제안하였다. Bob이 "홀수"라고 말하자 Alice는 그 번호가 Joe의 것임을 알려주고, Bob은 전화번호부에서 정답을 확인할 수 있다.

위의 예는 각단하지만 암호 시스템에 대한 전반적인 시나리오를 표현하고 있다. 안전하지 않은 통신 채널, 평문, 암호화문, 시드와 같은 개념이 잘 표현되어 있다. 현대의 암호화 기술은 좀더 복잡한 형태이지만 기본적으로 평문과 키를 입력받아 암화화하는 함수와 암호와 키를 받아 복호화하는 함수로 요약할 수 있다.

대칭키와 비대칭키

암호화 함수와 복호화 함수에 사용되는 키가 동일한 경우 이를 대칭키라고 하고, 각각의 함수에 사용되는 키가 서로 다른 경우를 비대칭키라고 한다. 대칭키보다는 비대칭키의 활용도가 높아 비대칭키를 활용한 암호화 기술이 많이 쓰인다. 비대칭키 쌍에서 하나의 키(개인키)는 소유자만 가지고, 다른 하나의 키(공개키)는 모두가 접근할 수 있도록 한다. 비대칭키를 사용한 대표적인 활용은 다음과 같다.

문서 암호화

문서의 내용을 다른 사람이 보지 못하게하고 해당 사용자만 볼 수 있게하려면 해당 사용자가 발급한 공개키로 암호화하면 이를 복호화할 수 있는 사람은 개인키를 가진 사람만이 볼 수 있다.

인증/전자서명

문서의 내용을 개인키로 암호화하면, 공개키를 가진 누구나 볼 수 있지만 개인키를 가진 사람만이 그 암호문을 만들 수 있으므로 문서 작성자를 인증하게 된다.

키저장소

개인키는 디지털 세상에서 자신을 나타내는 것이므로 보관에 주의해야 한다. 이를 위해 다양한 형태의 키저장소가 존재한다. 키저장소는 개인키를 가지고 있으며, 원칙적으로 외부로 전달되지 않고, 키를 사용해 할 수 있는 다양한 기능을 수행한 후 그 결과만을 전달하는 API라고 이해하면 좋다. 여기에서는 Android에서 제공되는 AndroidKeyStore라는 키저장소를 사용해서 개인키를 보관한다. 자세한 내용은 다음 링크를 확인할 수 있다.

- https://developer.android.com/training/articles/keystore.html

지문 인증

지문인증은 인증도구 중에 하나로 패스워드 인증을 대체할 수 있다. 본 모듈에서는 지문 인증 모듈을 통해 개인 인증을 수행하여 키저장소 접근 권한을 얻고 해당 개인키를 얻어오게 된다. 만일, 패스워드를 사용할 경우라면 패스워드 인증으로 키저장소에 저장된 키의 접근 권한을 얻어야 한다.

MVC 모델

인증 모듈은 지문 인증을 사용해서 키를 얻어오는 모델과 지문 인증을 위한 화면(DialogFragment)을 제공한다. 보통의 경우, 어플리케이션의 룩&필을 맞추기 위해 지문 인증 화면을 수정해야 하야만 하는 경우가 빈번하기 때문에 모델과 화면간의 의존성이 없도록 만들어져 있다. 또한, 지문 인증 과정의 예외 처리도 사용자 요구에 따라 달라지기 때문에 직접 변경이 가능하다.

프로그램

해당 모듈은 10여개의 파일로 구성된 비교적 간단한 프로그램이기 때문에 샘플을 통해 손쉽게 사용법을 익힐 수 있다. 또한, 동작 구조의 이해를 위해 소스도 함께 제공하고 있으므로 저장소에 접근하여 확인할 수 있다.

예제 프로그램 실행해보기

예제 프로그램은 키를 키저장소(keystore)에 생성하고 지문인증을 통해 키를 얻어오는 과정을 보여주는 안드로이드 어플리케이션이다. 예제 프로그램 전체는 MIT 라이센스이므로 자유롭게 수정이 가능하며, 필요에 따라 원래 소스에 반영(pull request)할 수 있다.

예제 프로그램 가져오기

예제 프로그램은 https://bitbucket.org/cloudwallet/android-fingerprint-keychain 에 호스팅되고 있다. 다음 명령을 통해 간단히 소스를 가져올 수 있다.

```
git clone https://blocko_bylee@bitbucket.org/cloudwallet/android-fingerprint-keychain.git
```

명령행이 익숙하지 않다면 Android Studio 메뉴에서 "File > New > Project from Version Control > Git"을 사용할 수 있다.

예제 프로그램 구성

지문 인증 모듈

안드로이드의 키 저장소와 지문 인증 과정을 손쉽게 사용할 수 있도록 API를 제공한다.

안드로이드 어플리케이션

지문 인증 모듈을 어떻게 사용하는지 보여주기 위한 예제이다.

예제 프로그램 실행

예제 프로그램의 프로젝트에는 junit 테스트 케이스와 android app을 실행할 수 있다. 예제 프로그램을 실행하기 위해서는 대상 프로그램을 app으로 선택하고 실행하면 된다.

예제 프로그램 분석

MainActivity

크게 개인키를 저장소에서 생성하는 부분과 인증을 시작하는 부분으로 나뉜다. 유심히 보아야할 부분은 authenticate 메쏘드이다.

큰 흐름은 mac을 생성(line2~4)한 후 인증 절차(FingerprintAuthenticationProcess)에 전달(line28)하고, 성공하면 해당 mac의 doFinal을 호출(line17)할 수 있게 된다. 이때, 인증 절차에서 발생하는 이벤트를 FingerprintAuthenticationCallback가 처리(line11~24)하도록 한다. 본 예제에서 FingerprintAuthenticationCallback는 인증 절차에 대한 이벤트와 FragmentDialog에서 발생하는 사용자 이벤트를 모두 처리하도록 구현되어있다.

FingerprintAuthenticationCallback

FingerprintAuthenticationCallback는 다음의 두 인터페이스를 구현하고 있다. 구현 내용은 간단히 3번의 인증 기회안에 인증을 진행하는 내용이다.

FingerprintAuthenticationHandler

인증 절차에서 발생하는 이벤트들을 처리하는 인터페이스이다. 다음의 상황에 따라 해당 메쏘드가 호출된다.

상황	메쏘드	진행 여부
인증 절차 시작	onStart(FingerprintAuthenticationProcess process)	Y

인증 절차 종료	onEnd(FingerprintAuthenticationProcess process)	N
인증 성공	onSuccess(FingerprintAuthenticationProcess process)	N
인증 실패	onFailure(FingerprintAuthenticationProcess process, CharSequence help)	Y
에러	onError(FingerprintAuthenticationProcess process, Throwable exception)	N

- 진행 여부는 해당 이벤트 이후 계속 지문 인식 상태를 유지하는지 여부이다.

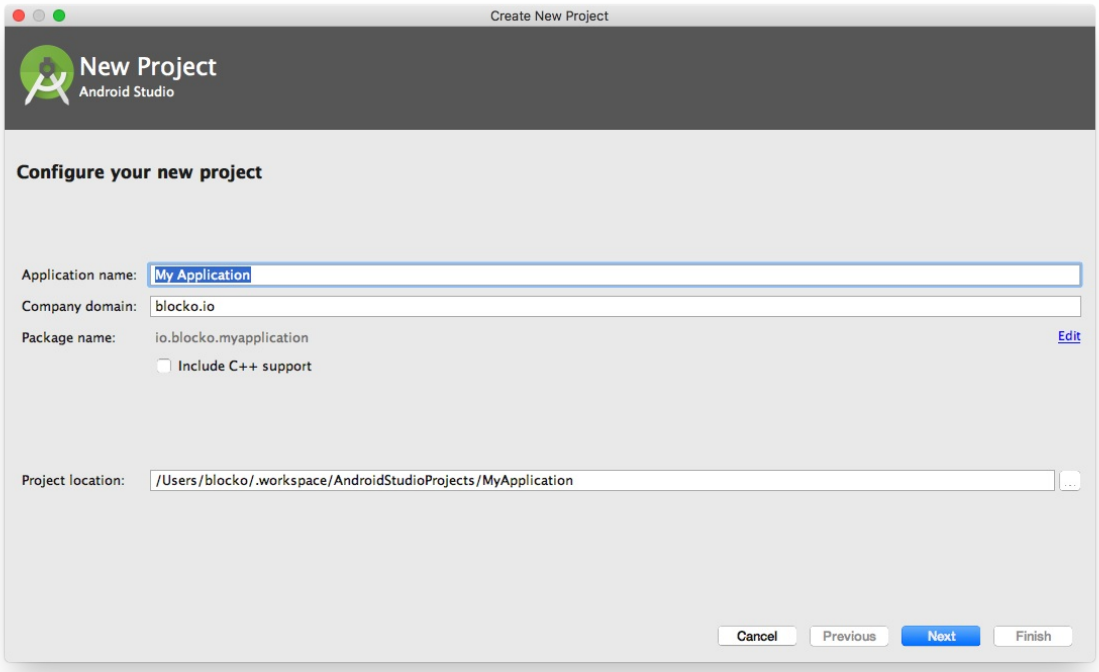
FingerprintAuthenticationDialogFragmentListener

FingerprintAuthenticationDialog가 dismiss 될때 onDismiss 메쏘드가 호출된다.

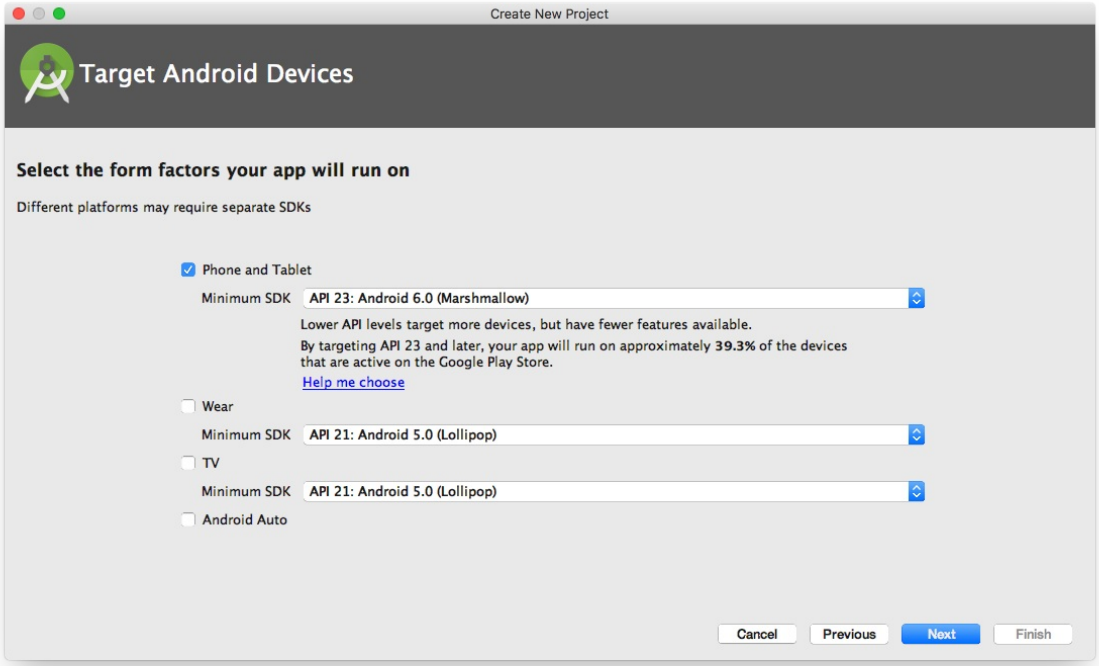
프로젝트 생성하기

Android Studio에서 프로젝트를 생성하는 것으로 부터 시작한다. 메뉴에서 "New > New Project..." 을 선택한다.

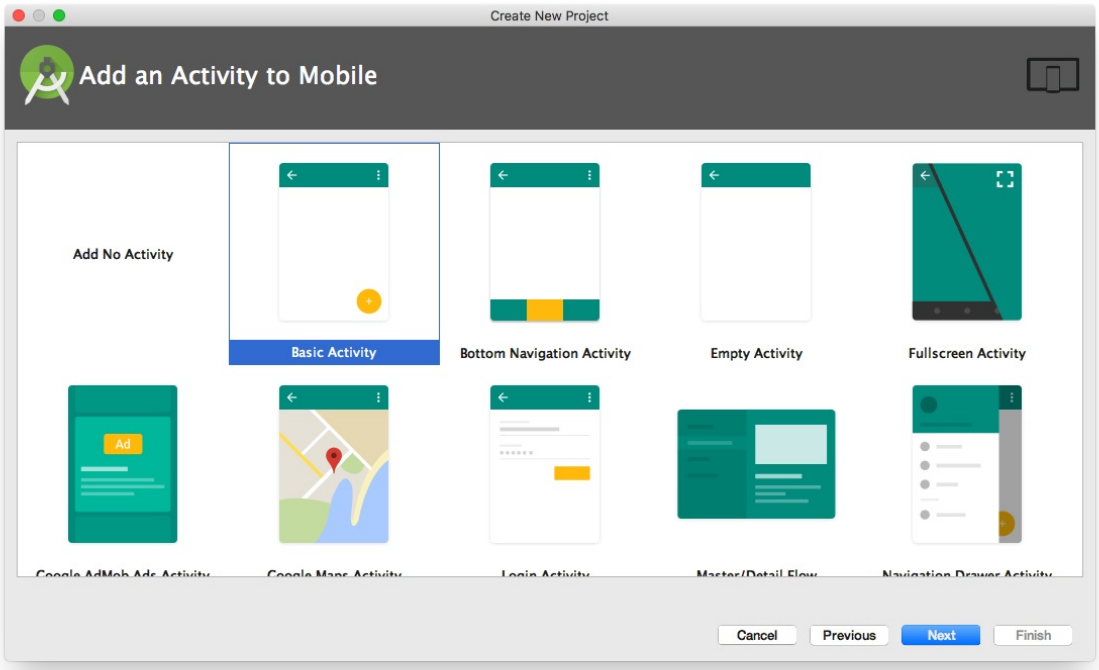
프로젝트 이름을 입력한다.



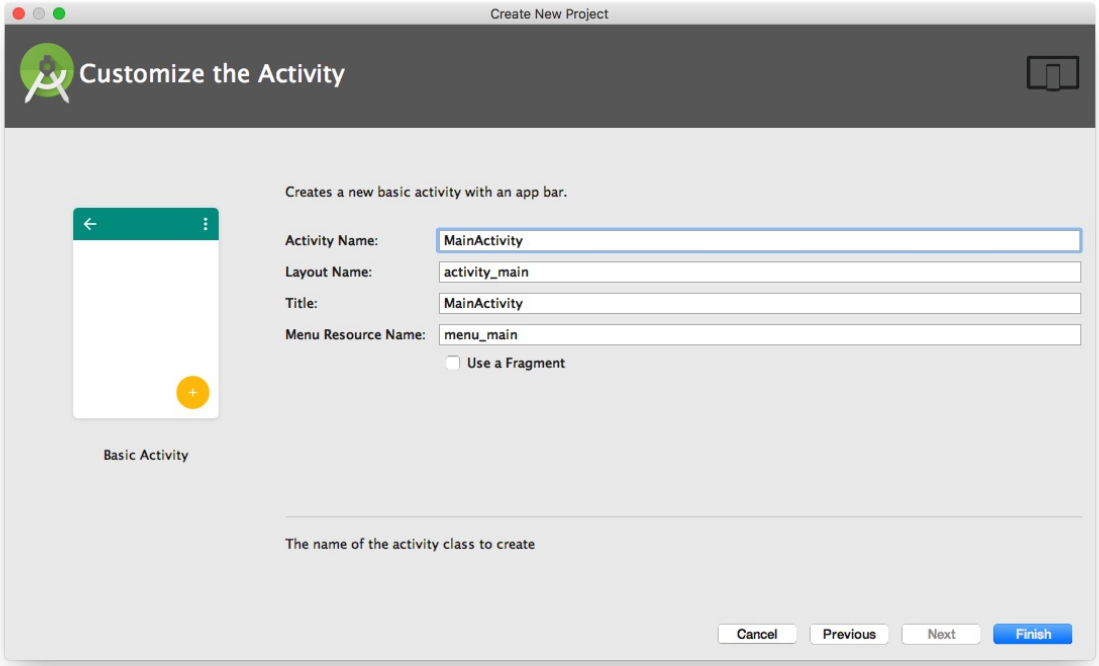
대상 플랫폼과 버젼을 선택한다.



생성할 앱 템플릿을 선택한다.



앱 템플릿에서 필요로 하는 이름등을 지정한다.



종료 버튼을 누르고 이상없이 프로젝트가 생성되었는지 확인한다.

Android Archive로 부터 설치하기

AAR(Android Archive)는 Android 라이브러리 모듈로써 어플리케이션에 포함되어 빌드되기 위한 일련의 정보를 포함하고 있다. 인증 모듈을 사용해서 개발을 진행하기 위해서는 Android Studio에서 AAR을 종속성에 추가해야 한다.

AAR 파일 가져오기

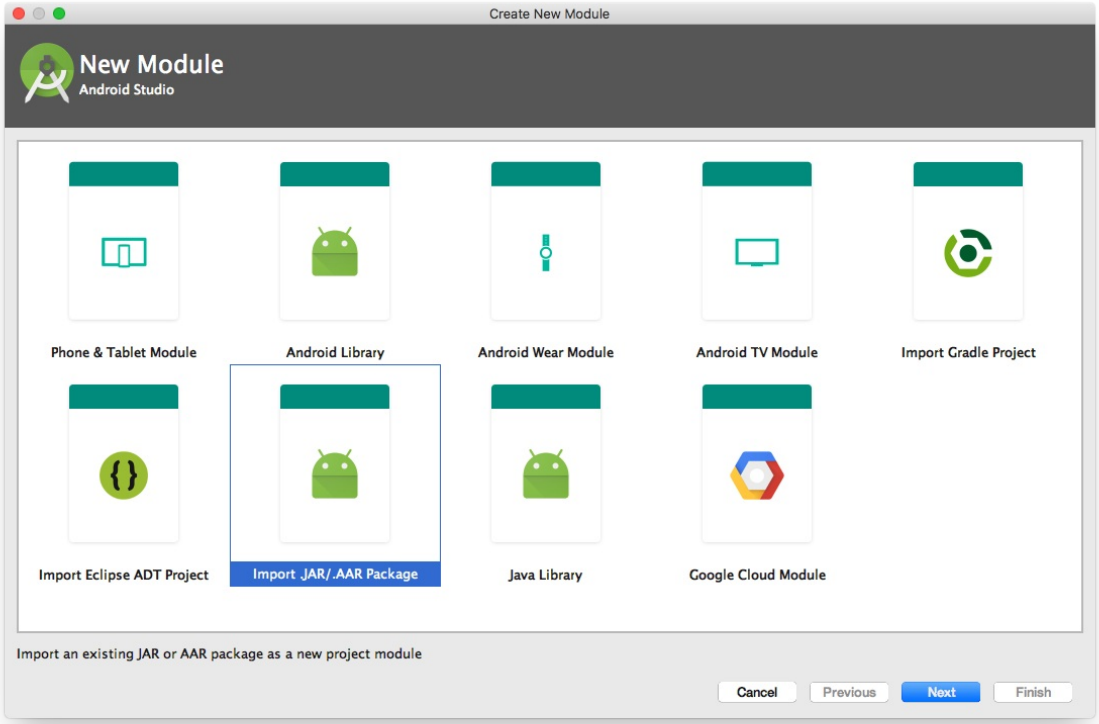
AAR 파일은 다음의 url로 부터 얻을 수 있다.

```
https://blocko.blob.core.windows.net/coinstack/coinstack-fingerprint-1.0.0.aar
```

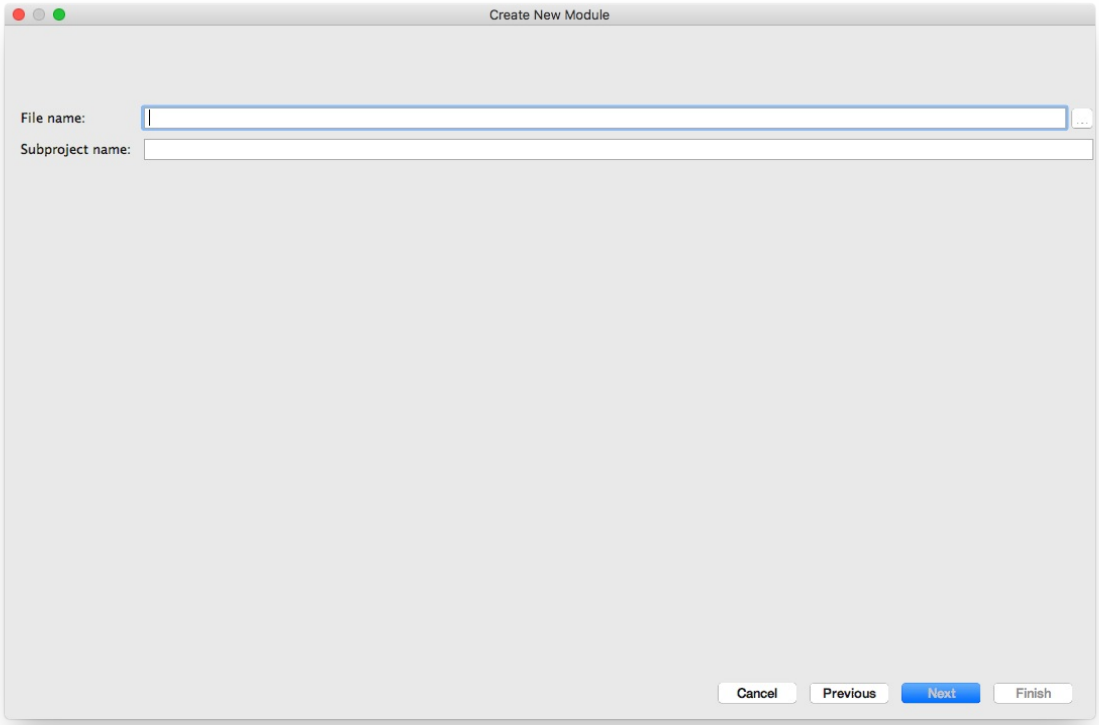
AAR 종속성에 추가하기

Android Studio에서 "File > New Module..."을 선택한다.

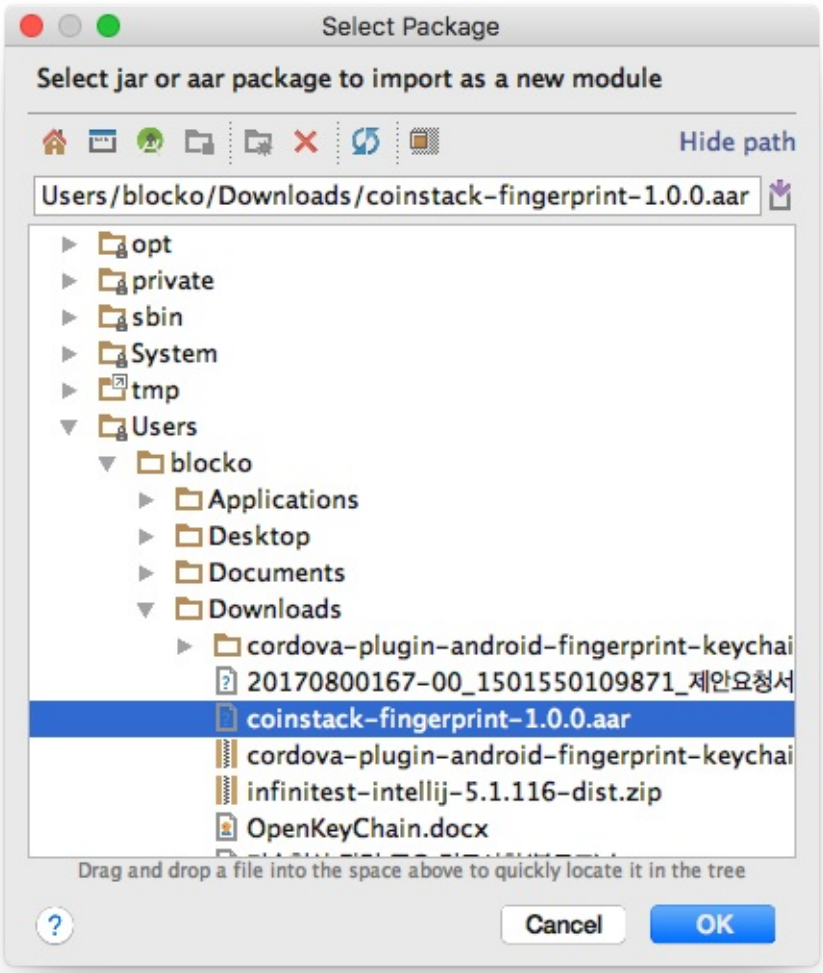
"Import .JAR / .AAR Package"를 선택한다.



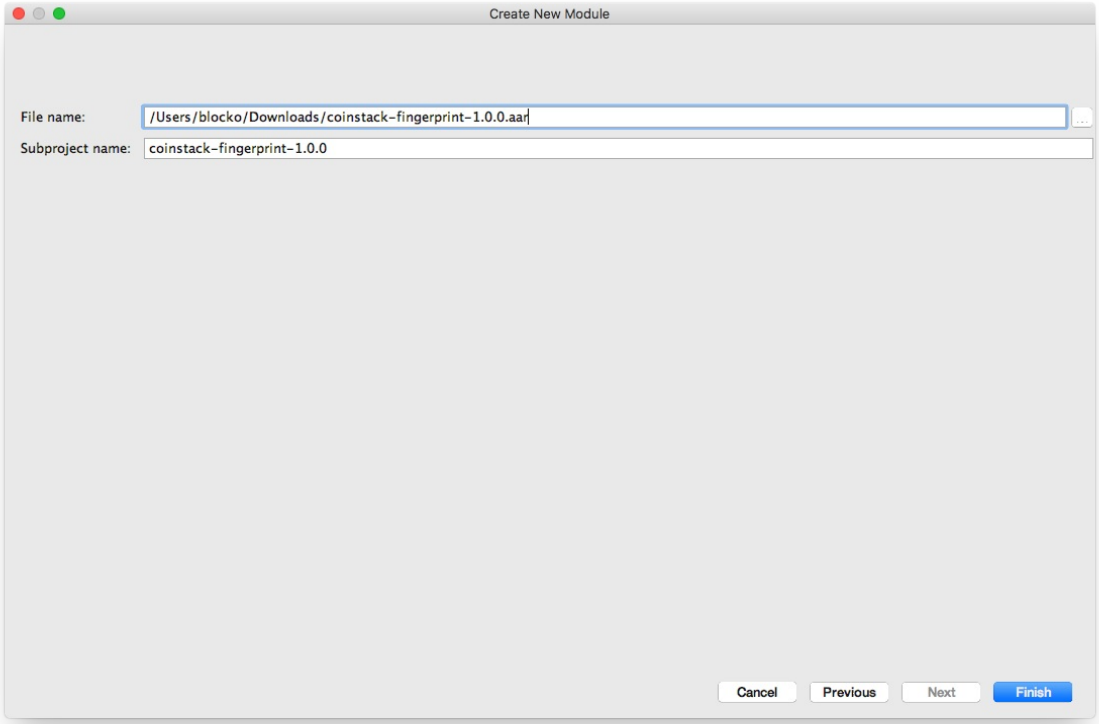
파일의 경로를 입력한다.



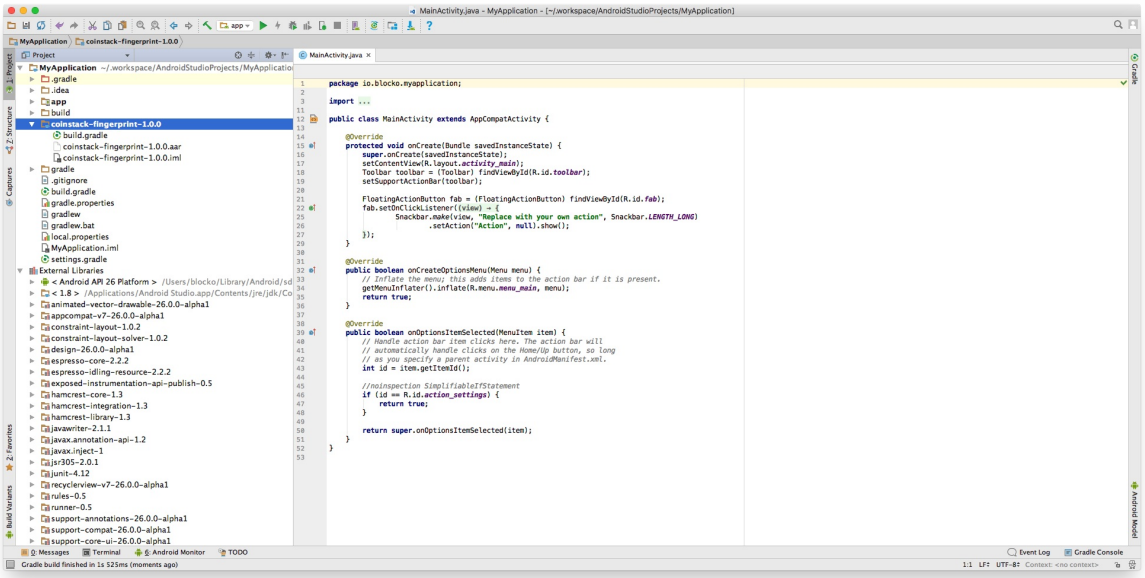
입력칸 우측에 버튼을 누르면 선택 대화창을 이용할 수 있다.



선택된 경로가 정상적으로 입력되었는지 확인한다.



종료 버튼을 누르고,프로젝트에 해당 모듈이 정상적으로 포함되었는지 확인한다.



FAQ

지문 인식 센서가 없는 경우

키를 KeyStore에 보관하면 키를 꺼내기 위해서는 안드로이드의 인증 API를 통해야만 KeyStore에 접근할 수 있다. 지문인증 모듈이 없는 경우 KeyStore의 접근권한을 패스워드나 패턴으로 획득하고, Key Store에 접근하면 된다.

Lock Out

일부 기기에서 Lock Out이 발생할 수 있는데, 이는 지극히 정상적인 현상이다. 구글은 안드로이드 제조업체에게 인증 수단(패스워드, 패턴그리기, 지문인증)에 관계없이 5회 연속 실패할 경우, 30초간 인증 시도를 하지 못하도록 가이드하고 있으며 이 기간을 Lock Out이라 부른다. Lock Out은 강력한 보안을 유지하기 위한 수단으로 LockOut이 발생할 경우, 이를 사용자에게 고지하고 Lock Out이 끝난후 인증을 시도하도록 가이드해야 한다. 자세한 사항은 아래의 문서를 참고한다.

https://source.android.com/compatibility/android-cdd.pdf