안드로이드 콜모듈

사용자 가이드

-V1.1.0-

목차

[1. 개요 3](#_Toc124946607)

[1.1 목적 3](#_Toc124946608)

[1.2 문서 연혁 3](#_Toc124946609)

[2 모듈 구조 3](#_Toc124946610)

[2.1 패키지 구조 3](#_Toc124946611)

[2.2 아키텍처 4](#_Toc124946612)

[3. 환경설정 5](#_Toc124946613)

[3.1 기본 앱으로 설정 5](#_Toc124946614)

[3.2 권한 설정 5](#_Toc124946615)

[3.3 InCallService 클래스 추가 5](#_Toc124946616)

[4. API 사용법 6](#_Toc124946617)

[4.1 콜 관련 API 6](#_Toc124946618)

[4.1.1 CallViewModel 6](#_Toc124946619)

[4.1.2 makeCall 6](#_Toc124946620)

[4.1.3 answerCall 6](#_Toc124946621)

[4.1.4 disconnectCall 6](#_Toc124946622)

[4.2 콜로그 관련 API 6](#_Toc124946623)

[4.2.1 CallLogViewModel 6](#_Toc124946624)

[4.2.2 getAllCallLog 6](#_Toc124946625)

[4.2.3 getCallLog 7](#_Toc124946626)

[4.2.4 deleteAllCallLog 7](#_Toc124946627)

[4.2.5 deleteCallLog 7](#_Toc124946628)

[4.3 연락처 관련 API 8](#_Toc124946629)

[4.3.1 ContactViewModel 8](#_Toc124946630)

[4.3.2 getContacts 8](#_Toc124946631)

[4.4 유틸 관련 API 8](#_Toc124946632)

[4.4.1 CallLogViewModel 8](#_Toc124946633)

[4.4.2 createDefaultDialerIntent 8](#_Toc124946634)

[5. 버전 정보 10](#_Toc124946635)

[6. 참고 자료 10](#_Toc124946636)

# 1. 개요

## 1.1 목적

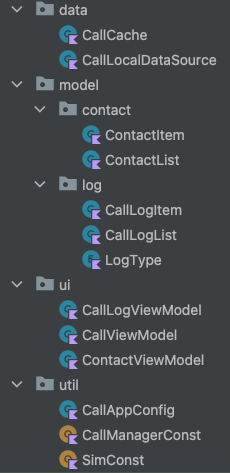
-본 문서는 안드로이드 디바이스의 콜 수신/발신 기능, 연락처 접근, 통화 기록 접근 기능을 사용할 수 있는  
 모듈의 특징과 사용법을 안내한다.

## 1.2 문서 연혁

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 버전 | 작성일자 | 내용 | 작성자 | 연락처 |
| 1.0.0 | 23.01.18 | 문서 초안 작성 | 박현준 | parkhj@konai.com |
| 1.1.0 | 23.01.26 | 4.1.2.2 추가 및 패키지 구조 수정 |
| 1.1.2 | 23.02.03 | 5. 버전 정보 추가 |

# 2 모듈 구조

## 2.1 패키지 구조

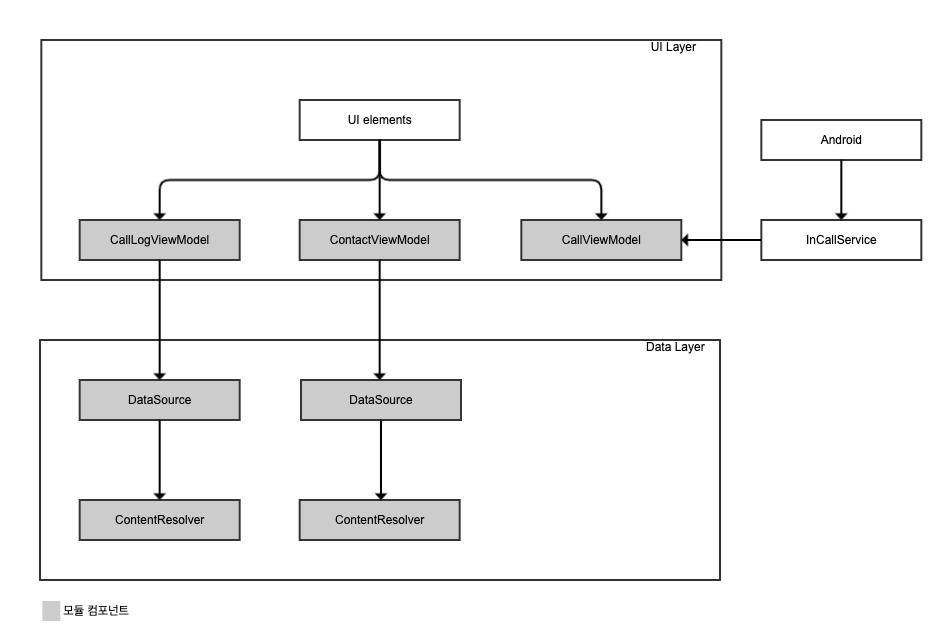


|  |  |  |
| --- | --- | --- |
| 레이어 | 패키지/클래스/오브젝트 | 내용 |
| Data (Layer) | data.model.contact | 연락처 정보를 관리하는 데이터 클래스 |
| data.model.log | 콜로그를 관리하는 데이터 클래스 |
| CallLocalDataSource | 컨텐트 리졸버에 접근해 콜로그, 연락처 데이터를 목적에 맞게 가공한다. |
| CallCache | 콜로그 관련 캐시 데이터 클래스(삭제 목록 생성 시 사용) |
| UI (Layer) | CallViewModel | 콜 객체를 관리하고, 콜 관련 사용자 인터페이스를 제공하는 클래스 |
| CallLogViewModel | 데이터 소스에서 콜로그를 가져오기 위한 사용자 인터페이스를  제공하고, 로그를 타입별로 분류해 LiveData로 관리하는 클래스 |
| ContactViewModel | 데이터 소스에서 연락처 데이터를 가져오기 위한 사용자 인터페이스를  제공하고, 해당 데이터를 LiveData로 관리하는 클래스 |
| - | CallManagerConst | 콜 관련 상수를 관리하는 오브젝트 |
| CallAppConfig | 모듈 사용 앱을 기본 앱으로 설정하는 인터페이스를 제공하는 클래스 |
| SimConst | 단말기 제조사 별 심 슬롯 이름을 관리하는 오브젝트 |

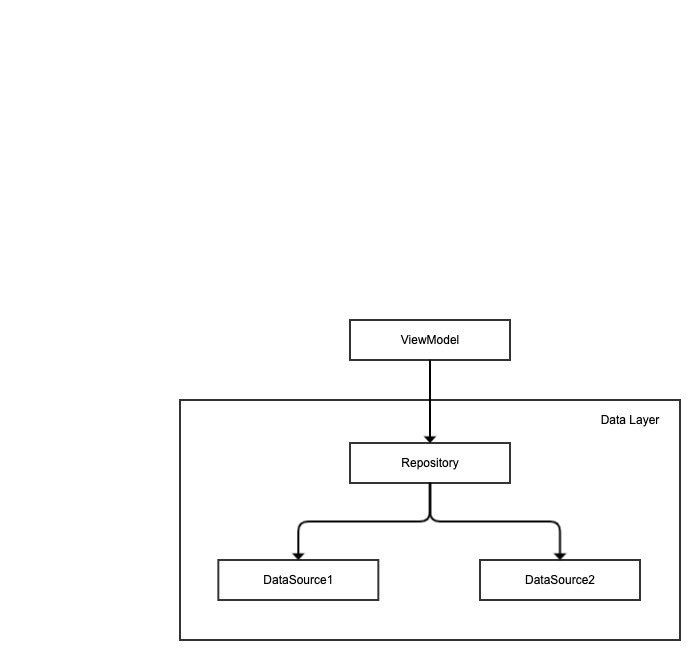
## 2.2 아키텍처

-해당 모듈은 MVVM 패턴으로 설계되었으며(그림1 참고), UI 구성 시 dataBinding 사용을 권장한다.

-목적에 따라 두개 이상의 데이터 소스에서 데이터 추출 후, 새로운 데이터를 생성 시 Repository 클래스 구현 후 사용을 권장한다.(그림 2 참고)



[그림 1 콜모듈 아키텍처]



[그림 2 Repository 클래스가 적용된 아키텍처]

# 3. 환경설정

## 3.1 기본 앱으로 설정

-해당 모듈의 콜 기능을 사용하기 위해서는 모듈을 사용하는 앱이 기본 전화 앱으로 설정되어 있어야한다.

-*‘4.4.2 createDefaultDialerIntent’* 참고

## 3.2 권한 설정

-모듈을 사용하기 위해서 필수 권한이 승인 되어야하며, AndroidManifest에 추가 후 권한을 허용한다.

|  |
| --- |
| **//AndroidManifest**  **<**uses**-**permission android:name**=**"android.permission.READ\_PHONE\_STATE" **/>**  **<**uses**-**permission android:name**=**"android.permission.READ\_CONTACTS" **/>**  **<**uses**-**permission android:name**=**"android.permission.CALL\_PHONE" **/>**  **<**uses**-**permission android:name**=**"android.permission.READ\_CALL\_LOG" **/>**  **<**uses**-**permission android:name**=**"android.permission.WRITE\_CALL\_LOG" **/>** |

## 3.3 InCallService 클래스 추가

-백그라운드 상태에서도 앱이 콜을 수신하기 위해서 *InCallService()*를 상속받은 클래스를 앱단에 추가하고, AndroidManifest <service>를 설정한다.

-*‘call.details.state’* API 31 이상 사용 가능

|  |
| --- |
| **Kotlin** |
| //App  class **CallService** : InCallService() {  override fun **onCallAdded**(call: Call) {  *//CallViewModel내부 call변수 초기화*  }  override fun **onCallRemoved**(call: Call) {  *//CallViewModel내부 call 변수 초기화*  }  }  //AndroidManifest  **<**service  android:name**=**".service.CallService"  android:permission**=**"android.permission.BIND\_INCALL\_SERVICE"  android:exported**=**"true"**>**  **<**meta**-**data  android:name**=**"android.telecom.IN\_CALL\_SERVICE\_UI"  android:value**=**"true" **/>**  **<**meta**-**data  android:name**=**"android.telecom.IN\_CALL\_SERVICE\_RINGING"  android:value**=**"true" **/>**  **<**intent**-**filter**>**  **<**action android:name**=**"android.telecom.InCallService"**/>**  **</**intent**-**filter**>**  **</**service**>** |

# 4. API 사용법

## 4.1 콜 관련 API

### 4.1.1 CallViewModel

4.1.1.1 Prototype

|  |
| --- |
| class **CallViewModel**(application: Application) : AndroidViewModel(application) |

4.1.1.2 Example

|  |
| --- |
| **Kotlin** |
| val callViewModel **=** CallViewModel(application) |

### 4.1.2 makeCall()

-기본 앱으로 전화 연결한다.

4.1.2.1 Prototype

|  |
| --- |
| fun **makeCall**(phoneNumber: String) |

4.1.2.2 Prototype

-eSIM 지원 단말기에서 전화 연결 시 사용하는 메서드로, USIM 과 eSIM 이 장착되어 있는 슬롯 인덱스를 파라미터로 전달한다.

|  |
| --- |
| fun **makeCall**(phoneNumber: String, simSlotIdx: Int) |

4.1.2.3 Example

|  |
| --- |
| **Kotlin** |
| //USIM 모델  callViewModel.makeCall("01012341234")  //eSIM 지원 모델  callViewModel.makeCall("01012341234", 1) |

### 4.1.3 answerCall()

-수신 전화를 받는다.

4.1.3.1 Prototype

|  |
| --- |
| fun **answerCall**() |

4.1.3.2 Example

|  |
| --- |
| **Kotlin** |
| callViewModel.answerCall() |

### 4.1.4 disconnectCall()

-전화 연결을 끊는다.

4.1.4.1 Prototype

|  |
| --- |
| fun **disconnectCall**() |

4.1.4.2 Example

|  |
| --- |
| **Kotlin** |
| callViewModel.disconnectCall() |

## 4.2 콜로그 관련 API

### 4.2.1 CallLogViewModel

4.2.1.1 Prototype

|  |
| --- |
| class **CallLogViewModel**(application: Application) : AndroidViewModel(application) |

4.2.1.2 Example

|  |
| --- |
| **Kotlin** |
| val callLogViewModel **=** CallLogViewModel(application) |

### 4.2.2 getAllCallLog()

-모든 콜로그(수신/발신/부재중)를 가져온다

4.2.2.1 Prototype

|  |
| --- |
| fun **getAllCallLog**() |

4.2.2.2 Example

|  |
| --- |
| **Kotlin** |
| *//val callLogList: LiveData<CallLogList> 초기화*  callLogViewModel.getAllCallLog()  *//result*  val allCallLog **=** callLogViewModel.callLogList.value |

### 4.2.3 getCallLog()

-수신/발신/부재중 로그 중 원하는 콜로그만 가져온다

4.2.3.1 Prototype

|  |
| --- |
| fun **getCallLog**(logType: LogType) |

4.2.3.2 Parameter

|  |
| --- |
| enum class **LogType**(val type: String) {  INCOMING("1"), *//수신*  OUTGOING("2"), *//발신*  MISSED("3"), *//부재중*  VOICEMAIL("4"),  REJECTED("5"),  BLOCKED("6"),  ANSWERED\_EXTERNALLY\_TYPE("7")  } |

4.2.2.3 Example

|  |
| --- |
| **Kotlin** |
| *//val outgoingCallLogList: LiveData<CallLogList> 초기화*  callLogViewModel.getCallLog(LogType.OUTGOING)  *//result*  val outgoingCallLog **=** callLogViewModel.outgoingCallLogList.value |

### 4.2.4 deleteAllCallLog()

-디바이스에 저장된 모든 콜로그를 삭제한다.

4.2.4.1 Prototype

|  |
| --- |
| fun **delateCallLog**() |

4.2.4.2 Example

|  |
| --- |
| **Kotlin** |
| callLogViewModel.delateCallLog() |

### 4.2.5 deleteCallLog()

-원하는 콜로그만 삭제한다.

4.2.5.1 Prototype

|  |
| --- |
| fun **deleteCallLog**(logIdList: ArrayList**<**String**>**) |

4.2.5.2 Example

|  |
| --- |
| **Kotlin** |
| *//삭제하려는 콜로그 식별자 리스트*  val logdIdList **=** arrayListOf("999" , "998")  callLogViewModel.deleteCallLog(logdIdList) |

## 4.3 연락처 관련 API

### 4.3.1 ContactViewModel

4.3.1.1 Prototype

|  |
| --- |
| class **ContactViewModel**(application: Application) : AndroidViewModel(application) |

4.3.1.2 Example

|  |
| --- |
| **Kotlin** |
| val contactViewModel **=** ContactViewModel(application) |

### 4.3.2 getContacts()

-기본 앱으로 전화 연결한다.

4.3.2.1 Prototype

|  |
| --- |
| fun **getContacts**() |

4.3.2.2 Example

|  |
| --- |
| **Kotlin** |
| *//val contactList: LiveData<ContactList> 초기화*  contactViewModel.getContacts()  *//result*  val contact **=** contactViewModel.contactList.value |

## 4.4 유틸 관련 API

### 4.4.1 CallLogViewModel

4.4.1.1 Prototype

|  |
| --- |
| class **CallUtil**(private val context: Context) |

4.4.1.2 Example

|  |
| --- |
| **Kotlin** |
| val callUtil **=** CallUtil(context) |

### 4.4.2 createDefaultDialerIntent()

-앱을 기본 전화 앱으로 설정한다.

-API 29 이상 사용 가능

4.4.2.1 Prototype

|  |
| --- |
| fun **createDefaultDialerIntent**(): Intent? |

4.4.2.2 Example

|  |
| --- |
| **Kotlin** |
| //App  fun **changeDefaultDialer**() {  val intent **=** callUtil.createDefaultDialerIntent()  changeDefaultDialerResultLauncher.launch(intent)  }    *//기본 전화 앱으로 변경 되었는지 결과를 확인하기 위한 ActivityResultLauncher*  private val changeDefaultDialerResultLauncher **=**  registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {  if (callUtil.isDefaultDialer) {  *//기본 앱으로 설정되었을 때*  } else {  *//기본 앱으로 설정되지 않았을 때*  }  } |

# 

# 5. 버전 정보

|  |  |
| --- | --- |
| 버전 | 기능 |
| 1.0.0 | 콜  -수신/발신/거절  연락처  -디바이스 저장 연락처 가져오기  최근 기록  -디바이스 전화 로그 가져오기  -로그 분류(수신/발신/부재중)  -로그 삭제하기  기타  -기본 앱 설정 |
| 1.1.0 | 콜  -eSIM 지원 단말기용 *makeCall(phoneNumber: String, simSlotIdx: Int)* 추가 |
| 1.1.1 | 콜  -CallManager – DataSource 를 CallLogLocalDataSource, ContactLocalDataSource 로 분리 |
| 1.1.2 | 심  -SimManager – UILayer, DataLayer 분리  버그  -권한 관련 버그 수정 및 디버깅용 로그 적용 |

# 6. 참고 자료

CallLog 안드로이드 공식 문서

https://developer.android.com/reference/android/provider/CallLog.Calls

Simple Phone GitHub

https://github.com/arekolek/simple-phone

InCallService로 Default Dialer 설정

https://greensky0026.tistory.com/207

기본 전화 앱 등록

https://raon-studio.tistory.com/13?category=1055922