
	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 1 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

Emkit Intent API Guide for SDK v3

Pointmobile Co., Ltd.
S/W R&D Team

COMPANY CONFIDENTIAL
Only to be disclosed to Point Mobile employees concerned with the project.
DO NOT MAKE COPIES

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 2 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

Emkit Intent API Guide for SDK v3

Revision History

Revision	Date	Change Description	Author
R01	2020.01.20	Initial draft	Ryan Kim
R02	2020.01.30	서식 및 표기 문구 수정	Emma Jung



	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 3 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

Table of Contents

1. Broadcast Intents	4
device.common.ENABLED_SCANNER.....	4
device.common.ENABLED_TRIGGER.....	4
device.common.ENABLED_TOUCHSCAN.....	5
device.common.READ_TOUCHSCAN_STATE	5
2. onReceive Intents.....	6
device.common.USERMSG	6
device.scanner.EVENT	7
device.common.STATE_TOUCHSCAN.....	8
3. AIM Identifier	9

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 4 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

1. Broadcast Intents

사용자 앱에서 전송하여 설정 가능한 인텐트들에 대한 설명이다.

device.common.ENABLED_SCANNER

바코드 스캐너에 대한 활성화/비활성화 설정을 할 수 있는 인텐트이다.

바코드 스캐너 상태는 단말기를 재시작하여도 유지되며 공장 초기 설정은 비활성화 상태이다.

(* Note : 설정 앱 - Settings > ScanSetting > Enable/Disable *)

Action	"device.common.ENABLED_SCANNER"
Parameter	Extra: "EXTRA_ENABLED_SCANNER" (int) 0 – Disabled 1 – Enabled
Example	<p>How to enable the scanner :</p> <pre>Intent scannerEnableIntent = new Intent("device.common.ENABLED_SCANNER"); scannerEnableIntent.putExtra("EXTRA_ENABLED_SCANNER", 1); sendBroadcast(scannerEnableIntent);</pre> <p>How to disable the scanner :</p> <pre>Intent scannerDisableIntent = new Intent("device.common.ENABLED_SCANNER"); scannerDisableIntent.putExtra("EXTRA_ENABLED_SCANNER", 0); sendBroadcast(scannerDisableIntent);</pre>

device.common.ENABLED_TRIGGER


바코드 스캐너가 활성화되어 있는 상태에서 트리거(바코드를 읽기 위해 빔이 나가는 동작) 동작에 대한 실행 가능/불가능 설정을 할 수 있는 인텐트이다.

주로 특정 순간에 바코드를 읽지 않도록 설정할 때 사용한다. (바코드 스캐너 자체를 활성화/비활성화는 시간이 오래 걸리는 단점이 있다.)

트리거 상태는 단말기를 재시작하면 항상 공장 초기 설정인 활성화 상태로 초기화된다.

(* Note : [Settings > ScanSetting] 에서 설정 불가능, API 또는 인텐트로만 설정 가능하다. *)

Action	"device.common.ENABLED_TRIGGER"
Parameter	Extra: "EXTRA_ENABLED_TRIGGER" (int) 0 – Disabled 1 – Enabled
Example	<p>How to enable the trigger :</p> <pre>Intent triggerEnableIntent = new Intent("device.common.ENABLED_TRIGGER"); triggerEnableIntent.putExtra("EXTRA_ENABLED_TRIGGER", 1); sendBroadcast(triggerEnableIntent);</pre> <p>How to disable the trigger :</p> <pre>Intent triggerDisableIntent = new Intent("device.common.ENABLED_TRIGGER"); triggerDisableIntent.putExtra("EXTRA_ENABLED_TRIGGER", 0); sendBroadcast(triggerDisableIntent);</pre>

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 5 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

device.common.ENABLED_TOUCHSCAN

단말기 화면에서 TouchScan 버튼(트리거 동작을 수행하는 버튼)을 활성화/비활성화 할 수 있는 인텐트이다.

바코드 스캐너가 활성화되어 있지 않으면, TouchScan 버튼을 활성화하여도 버튼 아이콘이 보이지 않고, 반대로 TouchScan 버튼이 활성화되어 있는 상태에서 바코드 스캐너를 비활성화하면 버튼 아이콘이 사라진다. 추가로 바코드 스캐너가 활성화되어 있지만 이미지 모드로 진입하면 TouchScan 버튼 아이콘이 사라진다.

TouchScan 버튼 활성화/비활성화 설정은 단말기를 재시작하여도 유지된다.

공장 초기 설정은 비활성화 상태이다.

(* Note : 설정 앱 - Settings > ScanSetting > Basic > Enable TouchScan *)

Action	"device.common.ENABLED_TOUCHSCAN"
Parameter	Extra: "EXTRA_ENABLED_TOUCHSCAN" (int) 0 – Disabled 1 – Enabled
Example	<p>How to enable the TouchScan :</p> <pre>Intent touchScanEnableIntent = new Intent("device.common.ENABLED_TOUCHSCAN"); touchScanEnableIntent.putExtra("EXTRA_ENABLED_TOUCHSCAN", 1); sendBroadcast(touchScanEnableIntent);</pre> <p>How to disable the TouchScan :</p> <pre>Intent touchScanDisableIntent = new Intent("device.common.ENABLED_TOUCHSCAN"); touchScanDisableIntent.putExtra("EXTRA_ENABLED_TOUCHSCAN", 0); sendBroadcast(touchScanDisableIntent);</pre>


device.common.READ_TOUCHSCAN_STATE

TouchScan 버튼 설정이 현재 활성화/비활성화 되어 있는지 확인할 수 있는 인텐트이다.

이 인텐트를 전송하면 device.common.STATE_TOUCHSCAN 인텐트를 통해 TouchScan 버튼의 상태를 받을 수 있다.

(* Note : 설정 앱 - Settings > ScanSetting > Basic > Enable TouchScan *)

Action	"device.common.READ_TOUCHSCAN_STATE"
Parameter	None
Example	<pre>Intent readTouchScanStateIntent = new Intent("device.common.READ_TOUCHSCAN_STATE"); sendBroadcast(readTouchScanStateIntent);</pre>

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 6 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

2. onReceive Intents

사용자 앱의 리시버로 전달받아 사용하는 인텐트들에 대한 설명이다.

device.common.USERMSG


바코드 디코드 완료 후 실패/성공에 대한 결과만 전달한다.

aDecodeGetResultType() 의 리턴 값이 DCD_RESULT_USERMSG 인 경우에만 동작한다.

(* Note : [Settings > ScanSetting > Basic > Wedge mode > Result type] = 'User Message' *)

사용자 앱에서는 인텐트에 대한 리시버를 만들어 주어야 하며, 추가로 디코드 결과를 세부적으로 알고 싶은 경우 aDecodeGetResult() 를 사용한다.

Action	"device.common.USERMSG"
Parameter	Extra: "EXTRA_USERMSG" (boolean) false – Decoding fail true – Decoding success
Example	<pre> public class ScanResultReceiver extends BroadcastReceiver { @Override public void onReceive(Context context, Intent intent) { if (mScanner != null) { if ("device.common.USERMSG".equals(intent.getAction())) { boolean isReadSuccess = intent.getBooleanExtra("EXTRA_USERMSG", false); if (isReadSuccess) { mScanner.aDecodeGetResult(mDecodeResult.recycle()); Log.d(TAG, "SCAN SUCCESS : " + mDecodeResult.toString()); } else { Log.d(TAG, "SCAN FAIL !!"); } } } } } </pre>

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 7 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

device.scanner.EVENT


디코드 완료 후 실패/성공에 대한 결과와 세부 데이터를 함께 전달한다.

aDecodeGetResultType()의 리턴 값이 RESULT_EVENT 인 경우에만 동작한다.

(* Note : [Settings > ScanSetting > Basic > Wedge mode > Result type] = 'Intent Broadcast' *)

사용자 앱에서는 아래 인텐트에 대한 리시버를 만들어 주어야 한다.

Action	"device.common.EVENT"
Parameter	<p>Extra: "EXTRA_EVENT_DECODE_RESULT" (boolean) false – Decoding fail true – Decoding success</p> <p>Extra: "EXTRA_EVENT_DECODE_LENGTH" (int) 디코드 데이터의 길이 ("EXTRA_EVENT_DECODE_VALUE"를 통해 얻을 수 있는 byte[] 길이와 동일하다.)</p> <p>Extra: "EXTRA_EVENT_DECODE_VALUE" (byte[]) 디코드 데이터 (byte[] 형식으로 되어 있고 주로 string 으로 변환하여 사용한다.)</p> <p>Extra: "EXTRA_EVENT_DECODE_LETTER" (byte) 디코드된 바코드에 대한 AIM identifier의 Code Character 값 (AIM identifier 항목 참조)</p> <p>Extra: "EXTRA_EVENT_DECODE_MODIFIER" (byte) 디코드된 바코드에 대한 AIM identifier의 Modifier Character 값 (AIM identifier 항목 참조)</p> <p>Extra: "EXTRA_EVENT_DECODE_TIME" (int) 트리거가 시작된 시점부터 디코드가 완료 되기까지 소요된 시간 (단위: ms)</p> <p>Extra: "EXTRA_EVENT_SYMBOL_NAME" (string) 디코드된 바코드 이름</p> <p>Extra: "EXTRA_EVENT_SYMBOL_ID" (byte) 디코드된 바코드 ID ([ScanSetting > Symbolologies > (Symbolology name) > Barcode ID] 값과 동일하다.)</p> <p>Extra: "EXTRA_EVENT_SYMBOL_TYPE" (int) 디코드된 바코드 종류 (ScanConst.SymbolologyID 에 정의된 값)</p>

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 8 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

Example	<pre> public class ScanResultReceiver extends BroadcastReceiver { @Override public void onReceive(Context context, Intent intent) { if (mScanner != null) { if ("device.scanner.EVENT".equals(intent.getAction())) { boolean result = intent.getBooleanExtra("EXTRA_EVENT_DECODE_RESULT", false); int decodeBytesLength = intent.getIntExtra("EXTRA_EVENT_DECODE_LENGTH", 0); byte[] decodeBytesValue = intent.getByteArrayExtra("EXTRA_EVENT_DECODE_VALUE"); String decodeValue = new String(decodeBytesValue, 0, decodeBytesLength); int decodeLength = decodeValue.length(); String symbolName = intent.getStringExtra("EXTRA_EVENT_SYMBOL_NAME"); byte symbolId = intent.getByteExtra("EXTRA_EVENT_SYMBOL_ID", (byte) 0); int symbolType = intent.getIntExtra("EXTRA_EVENT_SYMBOL_TYPE", 0); byte letter = intent.getByteExtra("EXTRA_EVENT_DECODE_LETTER", (byte) 0); byte modifier = intent.getByteExtra("EXTRA_EVENT_DECODE_MODIFIER", (byte) 0); int decodingTime = intent.getIntExtra("EXTRA_EVENT_DECODE_TIME", 0); Log.d(TAG, "1. result: " + result); Log.d(TAG, "2. bytes length: " + decodeBytesLength); Log.d(TAG, "3. bytes value: " + decodeBytesValue); Log.d(TAG, "4. decoding length: " + decodeLength); Log.d(TAG, "5. decoding value: " + decodeValue); Log.d(TAG, "6. symbol name: " + symbolName); Log.d(TAG, "7. symbol id: " + symbolId); Log.d(TAG, "8. symbol type: " + symbolType); Log.d(TAG, "9. decoding letter: " + letter); Log.d(TAG, "10. decoding modifier: " + modifier); Log.d(TAG, "11. decoding time: " + decodingTime); } } } } </pre>
---------	--


device.common.STATE_TOUCHSCAN

TouchScan 버튼의 활성화/비활성화 상태에 대한 인텐트이다.

사용자가 device.common.READ_TOUCHSCAN_STATE 인텐트를 전송했을 때에만 받을 수 있다.

사용자 앱에서는 아래 인텐트에 대한 리시버를 만들어 주어야 한다.

Action	"device.common.STATE_TOUCHSCAN"
Parameter	Extra: "EXTRA_TOUCHSCAN_STATE" (int) 0 – Disabled 1 – Enabled
Example	<pre> private class TouchScanReceiver extends BroadcastReceiver { @Override public void onReceive(Context context, Intent intent) { if ("device.common.STATE_TOUCHSCAN".equals(intent.getAction())) { int touchScanEnable = intent.getIntExtra("EXTRA_TOUCHSCAN_STATE", 0); if(touchScanEnable == 1) { Log.d(TAG, "TouchScan is enabled"); } else { Log.d(TAG, "TouchScan is disabled"); } } } } </pre>

	Doc.Type API Guide	Author SW R&D Dept.		Page of Pages 9 / 9
	Model PM Android Models	Doc.No.	Rev. R02	Date 2020.01.30

3. AIM Identifier

The AIM Code ID is a 3 character ISO/IEC identifier (originally created by AIM) generated by the decoder of a scanner, and gives information about the symbology of the barcode which was scanned.

Format	1 <i>cm</i>
Where	1 = flag character (ASCII 93) c = code character (defined by AIM for that symbology) m = modifier character