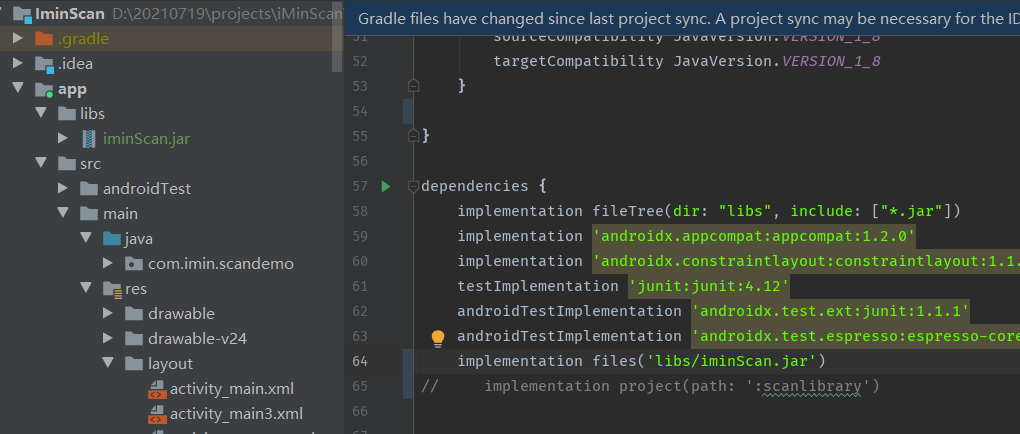
# IMIN扫码

###### 怎样使用imin的扫码sdk

1. 初始化响应配置，开发者的应用扫码类继承SDK的扫码界面CaptureActivity，初始化相关配置，获取返回值
2. 自己写相机界面，调用SDK完成图片解析，该方式相对复杂，但提供了更高的自由度

##### 首先在项目中引入imin扫码SDK，如下如所示：



##### 第一种使用方式：

为了降低开发难度，imin 扫码SDK封装了一个扫码模块。开发者需要在调用的界面继承CaptureActivity，然后进行扫码配置，在onScanResultCallback(Result result)方法中接收返回值

@SuppressLint("NewApi")

public class CustomCaptureActivity extends CaptureActivity {

StringBuilder sb = new StringBuilder();

private Toast toast;

private TextView textresult;

public int decode\_count = 0;

@Override

public int getLayoutId() {

return R.layout.custom\_capture\_activity;

}

@Override

public void onCreate(Bundle icicle) {

super.onCreate(icicle);

// Toolbar toolbar = findViewById(R.id.toolbar);

textresult = findViewById(R.id.textresult);

}

@Override

public void initCameraScan() {//初始化摄像头

super.initCameraScan();

//初始化解码配置

DecodeConfig decodeConfig = new DecodeConfig();

decodeConfig.setHints(DecodeFormatManager.ALL\_HINTS)////设置解码

.setSupportVerticalCode(true)//设置是否支持扫垂直的条码 （增强识别率，相应的也会增加性能消耗）

.setSupportLuminanceInvert(true)//设置是否支持识别反色码，黑白颜色反转（增强识别率，相应的也会增加性能消耗）

.setAreaRectRatio(0.8f)//设置识别区域比例，默认0.8，设置的比例最终会在预览区域裁剪基于此比例的一个矩形进行扫码识别

// .setAreaRectVerticalOffset(0)//设置识别区域垂直方向偏移量，默认为0，为0表示居中，可以为负数

// .setAreaRectHorizontalOffset(0)//设置识别区域水平方向偏移量，默认为0，为0表示居中，可以为负数

.setFullAreaScan(false);//设置是否全区域识别，默认false

//获取CameraScan，里面有扫码相关的配置设置。CameraScan里面包含部分支持链式调用的方法，即调用返回是CameraScan本身的一些配置建议在startCamera之前调用。

getCameraScan().setPlayBeep(true)//设置是否播放音效，默认为false

.setPlayRaw(R.raw.beep)

.setVibrate(true)//设置是否震动，默认为false

// .setCameraConfig(new CameraConfig())//设置相机配置信息，CameraConfig可覆写options方法自定义配置

.setCameraConfig(new ResolutionCameraConfig(CustomCaptureActivity.this))//设置CameraConfig，可以根据自己的需求去自定义配置

.setNeedAutoZoom(false)//二维码太小时可自动缩放，默认为false

.setNeedTouchZoom(false)//支持多指触摸捏合缩放，默认为true

.setDarkLightLux(45f)//设置光线足够暗的阈值（单位：lux），需要通过{@link #bindFlashlightView(View)}绑定手电筒才有效

.setBrightLightLux(100f)//设置光线足够明亮的阈值（单位：lux），需要通过{@link #bindFlashlightView(View)}绑定手电筒才有效

.bindFlashlightView(null)//绑定手电筒，绑定后可根据光线传感器，动态显示或隐藏手电筒按钮

.setOnScanResultCallback(this)//设置扫码结果回调，需要自己处理或者需要连扫时，可设置回调，自己去处理相关逻辑

.setAnalyzer(new MultiFormatAnalyzer(decodeConfig))//设置分析器,DecodeConfig可以配置一些解码时的配置信息，如果内置的不满足您的需求，你也可以自定义实现，

.setAnalyzeImage(true);//设置是否分析图片，默认为true。如果设置为false，相当于关闭了扫码识别功能

}

/\*\*

\* 扫码结果回调

\* @param result

\* @return 返回false表示不拦截，将关闭扫码界面并将结果返回给调用界面；

\* 返回true表示拦截，需自己处理逻辑。当isAnalyze为true时，默认会继续分析图像（也就是连扫）。

\* 如果只是想拦截扫码结果回调，并不想继续分析图像（不想连扫），请在拦截扫码逻辑处通过调

\* 用{@link CameraScan#setAnalyzeImage(boolean)}，

\* 因为{@link CameraScan#setAnalyzeImage(boolean)}方法能动态控制是否继续分析图像。

\*

\*/

@Override

public boolean onScanResultCallback(Result result) {

Resources resources = getResources();

sb.append(resources.getString(R.string.count) + decode\_count++);

sb.append("\n" + resources.getString(R.string.time\_consuming) + (System.currentTimeMillis()-result.getTimestamp()) + " ms\n");

sb.append(resources.getString(R.string.symbology) +result.getBarcodeFormat().name()+ "\n");

sb.append(resources.getString(R.string.capacity) + result.getNumBits()+ "\n");

sb.append(resources.getString(R.string.content) + result.toString());

textresult.setText(sb.toString());

sb.delete(0, sb.length());

/\*

\* 因为setAnalyzeImage方法能动态控制是否继续分析图像。

\*

\* 1. 因为分析图像默认为true，如果想支持连扫，返回true即可。

\* 当连扫的处理逻辑比较复杂时，请在处理逻辑前调用getCameraScan().setAnalyzeImage(false)，

\* 来停止分析图像，等逻辑处理完后再调用getCameraScan().setAnalyzeImage(true)来继续分析图像。

\*

\* 2. 如果只是想拦截扫码结果回调自己处理逻辑，但并不想继续分析图像（即不想连扫），可通过

\* 调用getCameraScan().setAnalyzeImage(false)来停止分析图像。

\*/

return true;

}

@Override

public void onScanResultFailure() {//扫描失败的回调

textresult.setText(sb.toString());

sb.delete(0, sb.length());

}

private void showToast(String text){

if(toast == null){

toast = Toast.makeText(this,text,Toast.LENGTH\_SHORT);

}else{

toast.setDuration(Toast.LENGTH\_SHORT);

toast.setText(text);

}

toast.show();

}

}

##### 第二种使用方式：

接入iminScan.jar后，

1.在处理业务的代码中引入相关java接口类

import com.google.zxing.Result;

import com.imin.scan.ScanUtils;

import com.imin.scan.Symbol;

2.相关参数初始化：

private ScanUtils scanUtils;

//获取解码工具类

scanUtils = ScanUtils.getInstance(this);

scanUtils.initScan();//初始化

scanUtils.initBeepSound(true,R.raw.beep);//初始化音频文件 可自定义扫码识别的音频文件

scanUtils.setConfig(Symbol.ALL\_FORMATS);//允许所有的解码类型

//详细解码类型

public interface Symbol {

int ALL\_FORMATS = 0;//所有 都支持

int ONE\_D\_FORMATS = 20; //一维码

int QR\_CODE\_FORMATS = 21;//二维码

int DATA\_MATRIX\_FORMATS = 22;//DATA\_MATRIX

int PRODUCT\_FORMATS = 23;//PRODUCT

/\*\* Aztec 2D barcode format. \*/

int AZTEC = 1;

/\*\* CODABAR 1D format. \*/

int CODABAR = 2;

/\*\* Code 39 1D format. \*/

int CODE\_39 = 3;

/\*\* Code 93 1D format. \*/

int CODE\_93 = 4;

/\*\* Code 128 1D format. \*/

int CODE\_128 = 5;

/\*\* Data Matrix 2D barcode format. \*/

int DATA\_MATRIX = 6;

/\*\* EAN-8 1D format. \*/

int EAN\_8 = 7;

/\*\* EAN-13 1D format. \*/

int EAN\_13 = 8;

/\*\* ITF (Interleaved Two of Five) 1D format. \*/

int ITF = 9;

/\*\* MaxiCode 2D barcode format. \*/

int MAXICODE = 10;

/\*\* PDF417 format. \*/

int PDF\_417 = 11;

/\*\* QR Code 2D barcode format. \*/

int QR\_CODE = 12;

/\*\* RSS 14 \*/

int RSS\_14 = 13;

/\*\* RSS EXPANDED \*/

int RSS\_EXPANDED = 14;

/\*\* UPC-A 1D format. \*/

int UPC\_A = 15;

/\*\* UPC-E 1D format. \*/

int UPC\_E = 16;

/\*\* UPC/EAN extension format. Not a stand-alone format. \*/

int UPC\_EAN\_EXTENSION = 17;

}

3.图像解码，如果是使用android系统相机解码，可以在图像预览回调函数即

PreviewCallback.onPreviewFrame(byte[] data, Camera camera)方法中直接调用预览数据解码

或者ImageReader.onImageAvailable(ImageReader reader)

方法中通过

Image image = reader.acquireLatestImage();

//我们可以将这帧数据转成字节数组，类似于Camera1的PreviewCallback回调的预览帧数据

ByteBuffer buffer = image.getPlanes()[0].getBuffer();

byte[] data = new byte[buffer.remaining()];

方法转换得到data原始数据进行解码

// 填充的数据data为摄像头原始数据 previewSize\_width,previewSize\_height 当前设定预览图的宽和高

Result rawResult = scanUtils.getScanResult(data,previewSize\_width,previewSize\_height);//

if (scanUtils.getNsyms() != 0 && rawResult != null) {//解码

}

4. 获取解码结果和条码类型。

if (scanUtils.getNsyms() != 0 && rawResult != null) {

Log.i("imin", "码型:"+rawResult.getBarcodeFormat().name());//条码类型,如“EAN-8”

Log.i("imin","结果:"+rawResult.toString())//解码结果字符串

}

更多说明

iMin扫码SDK目前支持的扫码类型包括如下：

* 一维码: EAN-8, EAN-13, UPC-A, UPC-E, CODABAR, Code39, Code93, Code128, RSS\_14, ITF, RSS\_EXPANDED，UPC\_EAN\_EXTENSION
* 二维码: QR Code , PDF417，DataMatrix，AZTEC，MAXICODE