

# QR Code 使用的套件及其類別

## 1 java.io

### File 類別

建立物件:

①File 物件 = new File( "路徑與檔名" )

②File 物件 = new File( "路徑", "檔名" )

## 2 java.nio.file

### Path 介面

建立物件:

①Path 物件 = Paths.get( "路徑與檔名" )

②Path 物件 = File 物件.toPath( )

註: java. nio. file. Paths

## 3 com.google.zxing.common

說明:是 Google 開發的套件

### (1)BitMatrix 類別

建立物件:

BitMatrix 物件 = MultiFormatWriter 物件.encode( String,  
format, 寬, 高, Map 物件)

註: ①format-**BarCodeFormat** 的屬性

## ②Map<EncodeHintType, Object>

Map 物件存入「編碼, 容錯率, 邊界」

## (2)HybridBinarizer 類別

建立物件:

```
HybridBinarizer 物件 = new HybridBinarizer( new  
    BufferedImageLuminanceSource(BufferedImage 物件))
```

註: HybridBinarizer 是 GlobalHistogramBinarizer 的子類別,  
用途是取得圖片的二維矩陣

## 4 com.google.zxing

說明:是 Google 開發的套件

## (1)EncodeHintType 類別

屬性:

①EncodeHintType.CHARACTER\_SET-編碼方式

②EncodeHintType.ERROR\_CORRECTION-設定容錯等級

註:ErrorCorrectionLevel 類別的常數-

❶L:7%

❷M:15% (default)

❸Q:25%

❹H:30%

(等級越高越不易識別)

③EncodeHintType.MARGIN-設留白大小

註: **0, 1(2), 3(4,5,6), 7(8,9,10)**

## (2)MultiFormatWriter 類別

建立物件:

MultiFormatWriter 物件 = new MultiFormatWriter( )

成員方法:

encode( 字串, Barcode 類型, 寬, 高, Map 物件 ): 建立

BitMatrix 物件

註: BitMatrix 物件是對應 QR\_Code 的二維矩陣

## (3)BarCodeFormat 類別

屬性:

①QR\_CODE

②CODE\_39-水電瓦斯...帳單(43 個文數字)

③CODE\_93-39 的升級版

④CODE\_128-身分證(可解碼 128 個 ASCII)

## (4)MultiFormatReader 類別

建立物件:

MultiFormatReader 物件 = new MultiFormatReader( )

成員方法:

decode(BinaryBitmap 物件, Map 物件):產生 Result 物件

## (5)BinaryBitmap 類別

建立物件:

```
BinaryBitmap 物件 = new BinaryBitmap ( HybridBinarizer 物件 )
```

## (6)Result 類別

建立物件:

```
Result 物件 = MultiFormatReader 物件.decode(  
  
BinaryBitmap 物件, Map 物件)
```

成員方法:

①getBarcodeFormat( ):取得二維碼格式

②getText( ):取得二維碼的內容

③toString( ):取得二維碼的內容

## 5 com.google. zxing.client.j2se

說明:是 Google 開發的套件

### (1)MatrixToImageWriter 類別

成員方法:

```
writeToPath(BitMatrix 物件, 格式, "路徑與檔名");
```

註:格式是指圖檔格式

❶jpg

❷png

## 6 java.awt.image

Javax.imageio

### BufferedImage 類別

建立物件:

① BufferedImage 物件 = new BufferedImage ( 寬, 高, type)

② BufferedImage 物件 = ImageIO.read( 路徑與檔名 )

### CreateQRCode.java

```
import java.io.File;
import java.nio.file.Path;
import java.util.HashMap;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.EncodeHintType;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.client.j2se.MatrixToImageWriter;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;

public class CreateQRCode
{
    public static void main(String[ ] args)
    {
        // 寬度和高度
        int width = 300;
        int height = 300;

        // 圖片的格式
        String format = "png";

        // 內容
```

```
String contents = "kuomulin,1019";

Path file = new File("./QR_CODE/name.png").toPath();

HashMap<EncodeHintType,Object> hints = new HashMap<EncodeHintType,Object>();

    hints.put(EncodeHintType.CHARACTER_SET, "utf-8");
    //容錯率且可以將它想像成解析度，分為 4 級：L(7%)，M(15%)，Q(25%)，H(30%)
    hints.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.M);
    hints.put(EncodeHintType.MARGIN, 2); //留白

    try {

        MultiFormatWriter writer = new MultiFormatWriter();
        BitMatrix bitMatrix = writer.encode(contents,
                                            BarcodeFormat.QR_CODE, width, height,hints);

        MatrixToImageWriter.writeToPath(bitMatrix, format, file);
        System.out.println("QR_Code 建立完成!!!");

    } catch (Exception e) {

        e.printStackTrace();
    }
}
```

## ReadQRCode.java

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.HashMap;
import javax.imageio.ImageIO;
```

```
import com.google.zxing.BinaryBitmap;
import com.google.zxing.EncodeHintType;
import com.google.zxing.MultiFormatReader;
import com.google.zxing.NotFoundException;
import com.google.zxing.Result;
import com.google.zxing.client.j2se.BufferedImageLuminanceSource;
import com.google.zxing.common.HybridBinarizer;

public class ReadQRCode
{
    public static void main(String[ ] args)
    {
        MultiFormatReader formatReader = new MultiFormatReader();

        File file = new File("./QR_CODE/name.png");

        BufferedImage image;
        try {

            image = ImageIO.read(file);
            BufferedImageLuminanceSource source = new
                BufferedImageLuminanceSource(image);

            BinaryBitmap binaryBitmap = new BinaryBitmap(
                new HybridBinarizer(source));

            HashMap hints = new HashMap();
            hints.put(EncodeHintType.CHARACTER_SET, "utf-8");

            // 解析

            Result result = formatReader.decode(binaryBitmap,hints);

            System.out.println("解析結果-");

            //System.out.println("格式 : "+result.getBarcodeFormat());

            System.out.println("內容 : "+result.getText());
```

```
        //System.out.println("内容：" + result.toString());  
    }  
    catch (IOException e) { }  
    }  
}
```