

```
1 takephoto.setOnClickListener(new OnClickListener() {
2     public void onClick(View v) {
3         File outputImage = new
File(getExternalCacheDir(), "output_image.jpg");
4         try {
5             if (outputImage.exists()) {
6                 outputImage.delete();
7             }
8             outputImage.createNewFile();
9         } catch (IOException e) {
10             e.printStackTrace();
11         }
12         imageUri =
FileProvider.getUriForFile(ContactEditor.this,
"com.example.ambow.contact.fileprovider",
outputImage);
13         Intent intent = new
Intent("android.media.action.IMAGE_CAPTURE");
14
15         intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
16         startActivityForResult(intent,
TAKE_PHOTO);
17     }
18     chooseFromAlbum.setOnClickListener(new
OnClickListener() {
19         public void onClick(View v) {
20             openAlbum();
21         }
22     });
23
24     private void openAlbum() {
25         Intent intent = new
Intent("android.intent.action.GET_CONTENT");
26         intent.setType("image/*");
```

```

27         startActivityForResult(intent,
    CHOOSE_PHOTO); // 打开相册
28     }
29
30     protected void onActivityResult (int
requestCode, int resultCode, Intent data){
31         switch (requestCode) {
32             case TAKE_PHOTO:
33                 if (resultCode == RESULT_OK) {
34                     try {
35                         // 将拍摄的照片显示出来
36                         Bitmap bitmap =
BitmapFactory.decodeStream(getContentResolver().openIn
putStream(imageUri));
37
38                         picture.setImageBitmap(bitmap);
39                     } catch (Exception e) {
40                         e.printStackTrace();
41                     }
42                     break;
43             case CHOOSE_PHOTO:
44                 if (resultCode == RESULT_OK) {
45                     // 判断手机系统版本号
46                     if (Build.VERSION.SDK_INT >=
19) {
47                         // 4.4及以上系统使用这个方法处
理图片
48                         handleImageOnKitKat(data);
49                     } else {
50                         // 4.4以下系统使用这个方法处理
图片
51                         handleImageBeforeKitKat(data);
52                     }
53                 }
54                 break;

```

```

55         default:
56             break;
57     }
58 }
59
60 private void handleImageOnKitKat(Intent data)
61 {
62     String imagePath = null;
63     Uri uri = data.getData();
64     Log.d("TAG", "handleImageOnKitKat: uri is
65 " + uri);
66     if (DocumentsContract.isDocumentUri(this,
67 uri)) {
68         // 如果是document类型的Uri, 则通过
69         document id处理
70         String docId =
71 DocumentsContract.getDocumentId(uri);
72         if
73 ("com.android.providers.media.documents".equals(uri.ge
74 tAuthority())) {
75             String id = docId.split(":")[1];
76             // 解析出数字格式 id
77             String selection =
78 MediaStore.Images.Media._ID + "=" + id;
79             imagePath =
80 getImagePath(MediaStore.Images.Media.EXTERNAL_CONTENT_
81 URI, selection);
82         } else if
83 ("com.android.providers.downloads.documents".equals(ur
84 i.getAuthority())) {
85             Uri contentUri =
86 ContentUris.withAppendedId(Uri.parse("content://downlo
87 ads/public_downloads"), Long.valueOf(docId));
88             imagePath =
89 getImagePath(contentUri, null);
90         }
91     }
92 }

```

```

75         } else if
76         ("content".equalsIgnoreCase(uri.getScheme())) {
77             // 如果是content类型的Uri, 则使用普通方式处
78             理
79             imagePath = getImagePath(uri, null);
80         } else if
81         ("file".equalsIgnoreCase(uri.getScheme())) {
82             // 如果是file类型的Uri, 直接获取图片路径即可
83             imagePath = uri.getPath();
84         }
85         displayImage(imagePath); // 根据图片路径显示
86         图片
87     }
88
89     private void handleImageBeforeKitKat (Intent
90     data){
91         Uri uri = data.getData();
92         String imagePath = getImagePath(uri,
93         null);
94         displayImage(imagePath);
95     }
96
97     private String getImagePath (Uri uri, String
98     selection){
99         String path = null;
100        // 通过Uri和selection来获取真实的图片路径
101        Cursor cursor =
102        getContentResolver().query(uri, null, selection, null,
103        null);
104        if (cursor != null) {
105            if (cursor.moveToFirst()) {
106                path =
107                cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
108            }
109            cursor.close();
110        }

```

```
101         return path;
102     }
103
104     private void displayImage (String imagePath){
105         if (imagePath != null) {
106             Bitmap bitmap =
107             BitmapFactory.decodeFile(imagePath);
108             picture.setImageBitmap(bitmap);
109         } else {
110             Toast.makeText(this, "failed to get
111             image", Toast.LENGTH_SHORT).show();
112         }
113     }
114 }
```