**Вывод изображений, различные способы**

Эта статья будет полезна начинающим разработчикам, здесь я предложу несколько вариантов вывода изображений на Android. Будут описаны следующие способы:  
  
**Обычный метод** – стандартный способ, используя ImageView. Рассмотрены варианты загрузки картинки из ресурса, а также из файла на SD карте устройства.  
  
**Продвинутый вариант** — вывод изображения, используя WebView. Добавляется поддержка масштабирования и прокрутки картинки при помощи жестов.  
  
**“Джедайский” способ** – улучшенный предыдущий вариант. Добавлен полноэкранный просмотр с автоматическим масштабированием изображения при показе и поддержкой смены ориентации устройства.  
  
Исходники тестового проекта на GitHub [github.com/Voldemar123/andriod-image-habrahabr-example](http://ajc.su/goto/https:/github.com/Voldemar123/andriod-image-habrahabr-example)  
  
В этой статье я не рассматриваю вопросы загрузки изображений из Интернета, кеширования, работы с файлами и необходимых для работы приложения permissions – только вывод картинок.  
  
Итак, задача — предположим, в нашем приложении необходимо вывести изображение на экран.  
Картинка может размерами превышать разрешение экрана и иметь различное соотношение сторон.   
Хранится она либо в ресурсах приложения, либо на External Storage — SD карте.  
  
Также допустим, мы уже записали на карту памяти несколько изображений (в тестовом проекте – загружаем из сети). Храним их в каталоге данных нашего приложения, в кеше.

|  |  |  |
| --- | --- | --- |
| 1 | public static final String APP\_PREFS\_NAME = Constants.class.getPackage().getName();<br> | |
|  |  |

|  |  |  |
| --- | --- | --- |
| 3 | public static final String APP\_CACHE\_PATH = <br> | |
| 4 |  |

|  |  |  |
| --- | --- | --- |
| 5 | Environment.getExternalStorageDirectory().getAbsolutePath() + <br> | |
| 6 |  |

|  |  |
| --- | --- |
| 7 | "/Android/data/" + APP\_PREFS\_NAME + "/cache/";<br> |

Для начала — обычный способ  
Официальная документация ImageView [developer.android.com/reference/android/widget/ImageView.html](http://ajc.su/goto/https:/developer.android.com/reference/android/widget/ImageView.html)   
  
Layout, где выводится картинка

|  |  |  |
| --- | --- | --- |
| 01 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"<br> | |
| 02 |  |

|  |  |  |
| --- | --- | --- |
| 03 | android:layout\_width="match\_parent"<br> | |
| 04 |  |

|  |  |  |
| --- | --- | --- |
| 05 | android:layout\_height="match\_parent"<br> | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | android:orientation="vertical" ><br><br> | |
| 08 |  |

|  |  |  |
| --- | --- | --- |
| 09 | <ImageView<br> | |
| 10 |  |

|  |  |  |
| --- | --- | --- |
| 11 | android:id="@+id/imageView1"<br> | |
| 12 |  |

|  |  |  |
| --- | --- | --- |
| 13 | android:layout\_width="fill\_parent"<br> | |
| 14 |  |

|  |  |
| --- | --- |
| 15 | android:layout\_height="fill\_parent" /><br><br> |

Масштабирование по умолчанию, по меньшей стoроне экрана.  
В Activity, где загружаем содержимое картинки

|  |  |  |
| --- | --- | --- |
| 1 | private ImageView mImageView;<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | mImageView = (ImageView) findViewById(R.id.imageView1);<br> |

Из ресурсов приложения (файл из res/drawable/img3.jpg)

|  |  |
| --- | --- |
| 1 | mImageView.setImageResource(R.drawable.img3);<br> |

Задавая Bitmap изображения

|  |  |  |
| --- | --- | --- |
| 1 | mImageView.setImageBitmap( imageUtil.getImageBitmap() );<br><br> | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | FileInputStream fis = new FileInputStream(Constants.APP\_CACHE\_PATH + this.image);<br> | |
| 4 |  |

|  |  |  |
| --- | --- | --- |
| 5 | BufferedInputStream bis = new BufferedInputStream(fis);<br><br> | |
| 6 |  |

|  |  |
| --- | --- |
| 7 | Bitmap img = BitmapFactory.decodeStream(bis);<br> |

Или передать URI на изображение (может хранится на карте или быть загружено из сети)

|  |  |  |
| --- | --- | --- |
| 1 | mImageView.setImageURI( imageUtil.getImageURI() );<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | Uri.fromFile( new File( Constants.APP\_CACHE\_PATH + this.image ) );<br> |

Этот способ стандартный, описан во множестве примеров и поэтому нам не особо интересен. Переходим к следующему варианту.  
  
Предположим, мы хотим показать большое изображение (например фотографию), которое размерами превышает разрешение нашего устройства. Необходимо добавить прокрутку и масштабирование картинки на экране.  
  
Для этого используем WebView   
Официальная документация WebView [developer.android.com/reference/android/webkit/WebView.html](http://ajc.su/goto/https:/developer.android.com/reference/android/webkit/WebView.html)

|  |  |  |
| --- | --- | --- |
| 01 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"<br> | |
| 02 |  |

|  |  |  |
| --- | --- | --- |
| 03 | android:layout\_width="match\_parent"<br> | |
| 04 |  |

|  |  |  |
| --- | --- | --- |
| 05 | android:layout\_height="match\_parent"<br> | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | android:orientation="vertical" ><br><br> | |
| 08 |  |

|  |  |  |
| --- | --- | --- |
| 09 | <WebView<br> | |
| 10 |  |

|  |  |  |
| --- | --- | --- |
| 11 | android:id="@+id/webView1"<br> | |
| 12 |  |

|  |  |  |
| --- | --- | --- |
| 13 | android:layout\_width="fill\_parent"<br> | |
| 14 |  |

|  |  |
| --- | --- |
| 15 | android:layout\_height="fill\_parent" /><br><br> |

В Activity, где загружаем содержимое

|  |  |  |
| --- | --- | --- |
| 1 | protected WebView webView;<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | webView = (WebView) findViewById(R.id.webView1);<br> |

установка черного цвета фона для комфортной работы (по умолчанию – белый)

|  |  |
| --- | --- |
| 1 | webView.setBackgroundColor(color.black);<br> |

включаем поддержку масштабирования

|  |  |  |
| --- | --- | --- |
| 1 | webView.getSettings().setSupportZoom(true);<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | webView.getSettings().setBuiltInZoomControls(true);<br> |

больше места для нашей картинки

|  |  |
| --- | --- |
| 1 | webView.setPadding(0, 0, 0, 0);<br> |

полосы прокрутки – внутри изображения, увеличение места для просмотра

|  |  |  |
| --- | --- | --- |
| 1 | webView.setScrollbarFadingEnabled(true);<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | webView.setScrollBarStyle(View.SCROLLBARS\_INSIDE\_OVERLAY);<br> |

загружаем изображение как ссылку на файл хранящийся на карте памяти

|  |  |  |
| --- | --- | --- |
| 1 | webView.loadUrl(imageUtil.getImageFileLink() );<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | "[file:///](NULL)" + Constants.APP\_CACHE\_PATH + this.image;<br> |

Теперь мы хотим сделать так, чтобы картинка при показе автоматически масштабировалась по одной из сторон, при этом прокрутка остается только в одном направлении.   
Например, для просмотра фотографий более удобна ландшафтная ориентация устройства.   
Также при смене ориентации телефона масштаб изображения должен автоматически меняться.   
Дополнительно расширим место для просмотра изображения на полный экран.  
  
В AndroidManifest.xml для нашей Activity добавляем

|  |  |  |
| --- | --- | --- |
| 1 | android:configChanges="orientation"<br> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen"<br> |

В код Activity добавлен метод, который вызыватся при каждом повороте нашего устройства.

|  |  |  |
| --- | --- | --- |
| 01 | <br> | |
| 02 |  |

|  |  |  |
| --- | --- | --- |
| 03 | @Override<br> | |
| 04 |  |

|  |  |  |
| --- | --- | --- |
| 05 | public void onConfigurationChanged(Configuration newConfig) {<br> | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | super.onConfigurationChanged(newConfig);<br> | |
| 08 |  |

|  |  |  |
| --- | --- | --- |
| 09 | changeContent();<br> | |
| 10 |  |

|  |  |
| --- | --- |
| 11 | } <br> |

В приватном методе описана логика пересчета масштаба для картинки  
Получаем информацию о размерах дисплея. Из-за того, что мы изменили тему Activity, теперь WebView раскрыт на полный экран, никакие другие элементы интерфейса не видны. Видимый размер дисплея равен разрешению экрана нашего Android устройства.

|  |  |  |
| --- | --- | --- |
| 1 | <br> | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | Display display = ((WindowManager) getSystemService(Context.WINDOW\_SERVICE)).getDefaultDisplay();<br><br> | |
| 4 |  |

|  |  |  |
| --- | --- | --- |
| 5 | int width = display.getWidth(); <br> | |
| 6 |  |

|  |  |
| --- | --- |
| 7 | int height = display.getHeight(); <br> |

Размеры изображения, выбранного для показа

|  |  |  |
| --- | --- | --- |
| 1 | <br> | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | Bitmap img = imageUtil.getImageBitmap();<br><br> | |
| 4 |  |

|  |  |  |
| --- | --- | --- |
| 5 | int picWidth = img.getWidth();<br> | |
| 6 |  |

|  |  |
| --- | --- |
| 7 | int picHeight = img.getHeight();<br> |

Меняем масштаб изображения если его высота больше высоты экрана. Прокрутка теперь будет только по горизонтали.

|  |  |  |
| --- | --- | --- |
| 01 | <br> | |
| 02 |  |

|  |  |  |
| --- | --- | --- |
| 03 | Double val = 1d;<br><br> | |
| 04 |  |

|  |  |  |
| --- | --- | --- |
| 05 | if (picHeight > height)<br> | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | val = new Double(height) / new Double(picHeight);<br><br> | |
| 08 |  |

|  |  |  |
| --- | --- | --- |
| 09 | val = val \* 100d;<br><br> | |
| 10 |  |

|  |  |
| --- | --- |
| 11 | webView.setInitialScale( val.intValue() );<br> |

Подбрасываем в WebView специально сформированный HTML файл, содержащий изображение.

[Смотреть исходники](http://ajc.su/mobilnye-os/android-mobilnye-os/android-vyvod-izobrazhenij-razlichnye-sposoby/#viewSource)

[печать](http://ajc.su/mobilnye-os/android-mobilnye-os/android-vyvod-izobrazhenij-razlichnye-sposoby/#printSource)[?](http://ajc.su/mobilnye-os/android-mobilnye-os/android-vyvod-izobrazhenij-razlichnye-sposoby/#about)

|  |  |  |
| --- | --- | --- |
| 01 | <br> | |
| 02 |  |

|  |  |  |
| --- | --- | --- |
| 03 | webView.loadDataWithBaseURL("/", <br> | |
| 04 |  |

|  |  |  |
| --- | --- | --- |
| 05 | imageUtil.getImageHtml(picWidth, picHeight), <br> | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | "text/html", <br> | |
| 08 |  |

|  |  |  |
| --- | --- | --- |
| 09 | "UTF-8", <br> | |
| 10 |  |

|  |  |  |
| --- | --- | --- |
| 11 | null);<br><br> | |
| 12 |  |

|  |  |  |
| --- | --- | --- |
| 13 | StringBuffer html = new StringBuffer();<br><br> | |
| 14 |  |

|  |  |  |
| --- | --- | --- |
| 15 | html.append("");<br> | |
| 16 |  |

|  |  |
| --- | --- |
| 17 | html.append("<img src="http://%22);%0D%0Ahtml.append(%[22file:///%22](http://ajc.su/mobilnye-os/android-mobilnye-os/android-vyvod-izobrazhenij-razlichnye-sposoby/%322file:/%22));%0D%0Ahtml.append(Constants.APP\_CACHE\_PATH);%0D%0Ahtml.append(this.image);%0D%0Ahtml.append(%22">");<br> |

Такой способ я применил из-того, что после загрузки изображения в WebView через метод loadUrl, как в прошлом варианте, setInitialScale после поворота устройства не изменяет масштаб картинки. Другими словами, показали картинку, повернули телефон, масштаб остался старый. Очень похоже на то, что изображение как-то кешируется.  
  
Я не нашел в документации упоминания об этом странном поведении. Может быть местные специалисты скажут, что я делаю не так?