Вывод изображений, различные способы

[Чулан](https://habrahabr.ru/hub/closet/)

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](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.  
  
public static final String APP\_PREFS\_NAME = Constants.class.getPackage().getName();  
public static final String APP\_CACHE\_PATH =   
Environment.getExternalStorageDirectory().getAbsolutePath() +   
"/Android/data/" + APP\_PREFS\_NAME + "/cache/";  
  
  
Для начала — обычный способ  
Официальная документация ImageView [developer.android.com/reference/android/widget/ImageView.html](https://developer.android.com/reference/android/widget/ImageView.html)   
  
Layout, где выводится картинка  
  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:orientation="vertical" >  
  
<ImageView  
android:id="@+id/imageView1"  
android:layout\_width="fill\_parent"  
android:layout\_height="fill\_parent" />  
  
  
  
  
Масштабирование по умолчанию, по меньшей стoроне экрана.  
В Activity, где загружаем содержимое картинки  
  
private ImageView mImageView;  
mImageView = (ImageView) findViewById(R.id.imageView1);  
  
  
Из ресурсов приложения (файл из res/drawable/img3.jpg)  
  
mImageView.setImageResource(R.drawable.img3);  
  
  
Задавая Bitmap изображения  
  
mImageView.setImageBitmap( imageUtil.getImageBitmap() );  
  
FileInputStream fis = new FileInputStream(Constants.APP\_CACHE\_PATH + this.image);  
BufferedInputStream bis = new BufferedInputStream(fis);  
  
Bitmap img = BitmapFactory.decodeStream(bis);  
  
  
Или передать URI на изображение (может хранится на карте или быть загружено из сети)  
  
mImageView.setImageURI( imageUtil.getImageURI() );  
Uri.fromFile( new File( Constants.APP\_CACHE\_PATH + this.image ) );  
  
  
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](https://developer.android.com/reference/android/webkit/WebView.html)  
  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:orientation="vertical" >  
  
<WebView  
android:id="@+id/webView1"  
android:layout\_width="fill\_parent"  
android:layout\_height="fill\_parent" />  
  
  
  
  
В Activity, где загружаем содержимое  
  
protected WebView webView;  
webView = (WebView) findViewById(R.id.webView1);  
  
  
установка черного цвета фона для комфортной работы (по умолчанию – белый)  
  
webView.setBackgroundColor(color.black);  
  
  
включаем поддержку масштабирования  
  
webView.getSettings().setSupportZoom(true);  
webView.getSettings().setBuiltInZoomControls(true);  
  
  
больше места для нашей картинки  
  
webView.setPadding(0, 0, 0, 0);  
  
  
полосы прокрутки – внутри изображения, увеличение места для просмотра   
  
webView.setScrollbarFadingEnabled(true);  
webView.setScrollBarStyle(View.SCROLLBARS\_INSIDE\_OVERLAY);  
  
  
загружаем изображение как ссылку на файл хранящийся на карте памяти  
  
webView.loadUrl(imageUtil.getImageFileLink() );  
"file:///" + Constants.APP\_CACHE\_PATH + this.image;  
  
  
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  
  
android:configChanges="orientation"  
android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen"  
  
  
В код Activity добавлен метод, который вызыватся при каждом повороте нашего устройства.  
  
@Override  
public void onConfigurationChanged(Configuration newConfig) {  
super.onConfigurationChanged(newConfig);  
changeContent();  
}   
  
  
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.  
  
  
Display display = ((WindowManager) getSystemService(Context.WINDOW\_SERVICE)).getDefaultDisplay();  
  
int width = display.getWidth();   
int height = display.getHeight();   
  
  
Размеры изображения, выбранного для показа  
  
  
Bitmap img = imageUtil.getImageBitmap();  
  
int picWidth = img.getWidth();  
int picHeight = img.getHeight();  
  
  
Меняем масштаб изображения если его высота больше высоты экрана. Прокрутка теперь будет только по горизонтали.  
  
  
Double val = 1d;  
  
if (picHeight > height)  
val = new Double(height) / new Double(picHeight);  
  
val = val \* 100d;  
  
webView.setInitialScale( val.intValue() );  
  
  
Подбрасываем в WebView специально сформированный HTML файл, содержащий изображение.  
  
  
webView.loadDataWithBaseURL("/",   
imageUtil.getImageHtml(picWidth, picHeight),   
"text/html",   
"UTF-8",   
null);  
  
StringBuffer html = new StringBuffer();  
  
html.append("");  
html.append("");  
  
  
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?