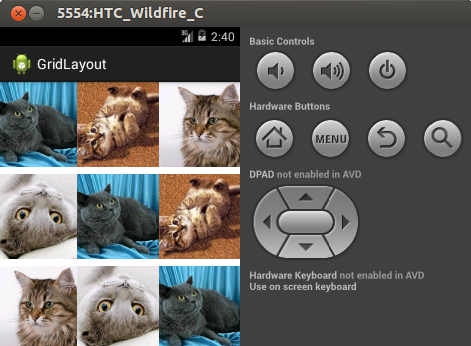
GridView — сетка в разметке экрана

[admin](http://androiddocs.ru/author/admin/) | 11.06.2014

[0 Comment](http://androiddocs.ru/gridview-setka-v-razmetke-ekrana/#comments)

GridView — это Layout, имеющий вид сетки или таблицы. Чаще всего используется для структурированного расположения изображений, например, при отображении галереи. В этом уроке мы создадим простой GridLayout, который будет заполняться изображениями из ресурсов с помощью ImageAdapter, который мы создадим. После запуска приложения должно получиться примерно такое:



Для начала подготовьте изображения. Скачайте из интернета любые картинки и поместите ее в папку**res/drawable** вашего проекта. Если ADT «не видит» новые картинки, кликните правой кнопкой на папке**drawable** и выберите **refresh**. В моем примере — 4 картинки с именами а2, а3, а4 и а5.

Создадим макет с разметкой GridView.

**res/layout/activity\_main.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <?xml version="1.0" encoding="utf-8"?>  <GridView xmlns:android="http://schemas.android.com/apk/res/android"  android:id="@+id/gridview"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:columnWidth="90dp"  android:numColumns="auto\_fit"  android:verticalSpacing="10dp"  android:horizontalSpacing="10dp"  android:stretchMode="columnWidth"  android:gravity="center"  /> |

Атрибут **android:numColumns** имеет значение **«auto\_fit»**, что означает, что количество колонок будет определяться автоматически в зависимости от размеров экрана. Ширина колонки 90dp.

Файл **MainActivity.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | package ru.androiddocs.gridlayout;    import adapters.ImageAdapter;  import android.support.v7.app.ActionBarActivity;  import android.widget.AdapterView;  import android.widget.AdapterView.OnItemClickListener;  import android.widget.GridView;  import android.widget.Toast;  import android.os.Bundle;  import android.view.View;      public class MainActivity extends ActionBarActivity {    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);    GridView grView = (GridView) findViewById(R.id.gridview);  grView.setAdapter(new ImageAdapter(this));    grView.setOnItemClickListener(new OnItemClickListener() {  public void onItemClick(AdapterView<?> parent, View v, int position, long id) {  Toast.makeText(MainActivity.this, "" + position, Toast.LENGTH\_SHORT).show();  }  });  }  } |

Здесь не так много кода, поэтому, думаю, разобраться будет несложно. Мы находим в layout наш GridView:

|  |  |
| --- | --- |
| 1 | GridView grView = (GridView) findViewById(R.id.gridview); |

и затем задаем адаптер:

|  |  |
| --- | --- |
| 1 | grView.setAdapter(new ImageAdapter(this)); |

и назначаем обработчик клика по картинке, который выводим с помощью [Toast](http://androiddocs.ru/vsplyvayushhie-soobshheniya-toast/) позицию изображения.

Создадим наш ImageAdapter, который будет наследовать базовый класс **BaseAdapter**. Я создал новую папку **adapters** в **src** и поместил новый класс туда.

**ImageAdapter.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | package adapters;    import ru.androiddocs.gridlayout.R;  import android.content.Context;  import android.view.View;  import android.view.ViewGroup;  import android.widget.BaseAdapter;  import android.widget.GridView;  import android.widget.ImageView;    public class ImageAdapter extends BaseAdapter {  private Context mContext;    public ImageAdapter(Context c) {  mContext = c;  }    public int getCount() {  return mThumbIds.length;  }    public Object getItem(int position) {  return null;  }    public long getItemId(int position) {  return 0;  }    public View getView(int position, View convertView, ViewGroup parent) {  ImageView imageView;  if (convertView == null) {  imageView = new ImageView(mContext);  imageView.setLayoutParams(new GridView.LayoutParams(85, 85));  imageView.setScaleType(ImageView.ScaleType.CENTER\_CROP);  } else {  imageView = (ImageView) convertView;  }    imageView.setImageResource(mThumbIds[position]);  return imageView;  }    private Integer[] mThumbIds = {  R.drawable.a2, R.drawable.a3,  R.drawable.a4, R.drawable.a5,  R.drawable.a2, R.drawable.a3,  R.drawable.a4, R.drawable.a5,  R.drawable.a2, R.drawable.a3,  R.drawable.a4, R.drawable.a5,  R.drawable.a2, R.drawable.a3,  R.drawable.a4, R.drawable.a5,  R.drawable.a2, R.drawable.a3,  R.drawable.a4, R.drawable.a5  };    } |

Как видите, тут есть несколько методов. Список картинок для отображения мы задали в приватном массиве **mThumbIdоs**. Тут просто повторяются 4 мои картинки. Можно задать столько изображений, сколько вам нужно.

Метод **getCount()** возвращает количество элементов массива с картинками. Именно столько раз будет вызван метод **getView()** для каждого изображения.

Методы getItem() и getItemId() для нас тут роли не играют, поэтому возвращают null и 0 (мы должны были добавить для них реализацию).