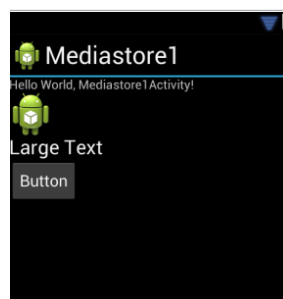


# MediaStore (ContextProvider)

## Bài 1. Đọc hình ảnh tuần tự từ thẻ nhớ.



1. Tạo project thiết kế giao diện gồm 1 ImageView, 1 TextView và 1 Button.



2. Trong file java khai báo và ánh xạ 3 đối tượng vừa tạo.
3. Khai báo thêm 3 biến là "soluong" để lưu tổng số hình, "mang" để lưu các đường dẫn đến hình.

```
int soluong;  
int index=0;  
ArrayList<String> mang=new ArrayList<String>();
```

4. Lấy dữ liệu trong ContentProvider bằng cách tạo ra đối tượng ContentResolver. Sau đó dùng query để truy vấn trong MediaStore provider và trả về con trỏ. Gọi getCount() để lấy ra số lượng.

```
ContentResolver ctr=this.getContentResolver();  
Cursor contro=ctr.query(MediaStore.Images.Thumbnails.EXTERNAL_CONTENT_URI,  
    null, null, null, null);  
soluong=contro.getCount();
```

5. Ta đã có con trỏ trỏ đến Provider chứa URI đến các hình ảnh. Tiếp theo ta cho con trỏ move đến vị trí đầu tiên. Trên từng hàng của dữ liệu ta phải chỉ ra là sẽ lấy dữ liệu trên cột nào nên tạo 1 biến tên "cotthu" và gọi hàm getColumnIndex và chỉ cột cần lấy là "DATA", sau đó dùng getString để lấy dữ liệu (đó chính là đường dẫn dẫn đến hình ảnh).

```
contro.moveToFirst();
for(int i=0;i<soluong;i++)
{
    int cotthu=contro.getColumnIndex(MediaStore.Images.Thumbnails.DATA);
    String src=contro.getString(cotthu);
    mang.add(src);
    contro.moveToNext();
}
contro.close();
```

6. Bắt sự kiện click chuột để lấy tuần tự từng Uri hình ra đổ lên ImageView và TextView.

```
bt.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        String src=mang.get(index);
        iv.setImageURI(Uri.parse(src));
        tv.setText(src);
        index++;
        if(index==soluong)
            index=0;
    }

});
```

## Bài 2. load hình ảnh từ thẻ nhớ lên GridView.

1. Tạo project tên MediaStore2, trên giao diện chính kéo vào 1 GridView.
2. Trong file java khởi tạo GridView, ánh xạ gridView.
3. Khai báo thêm 1 Cursor và một mảng chuỗi chỉ ra các cột nào trong đối tượng MediaStore mà ta cần lấy dữ liệu.

```
GridView gv;
Cursor cursor;
String[] cots={MediaStore.Images.Thumbnails._ID,MediaStore.Images.Thumbnails.DATA};
```

4. Trong onCreate, tạo đối tượng ContextResolver và gọi phương thức query để truy vấn đến URI của dữ liệu hình và trả về con trỏ.

```
ContentResolver ctr=this.getContentResolver();
cursor=ctr.query(
    MediaStore.Images.Thumbnails.EXTERNAL_CONTENT_URI, //url
    cots, //lay du lieu tu nhung cot nao (mang chuoai), null lay het
    null, //giong menh de where
    null, //menh de where dung ?s , cho bien, o day chua danh sach bien
    null); //sap xep theo cai gi
```

5. Xây dựng 1 class mới tên myadapter kế thừa từ BaseAdapter, xây dựng hàm tạo để lấy context và các hàm khác như getCount, getItem....

```
public class myadapter extends BaseAdapter{

    Context context;
    public myadapter(Context c){
        this.context=c;
    }
    public int getCount() {
        // TODO Auto-generated method stub
        return cursor.getCount();
    }
    public Object getItem(int arg0) {
        // TODO Auto-generated method stub
        return arg0;
    }
    public long getItemId(int arg0) {
        // TODO Auto-generated method stub
        return arg0;
    }
}
```

6. Xây dựng hàm getView cho lớp myAdapter như sau, tạo ra 1 imageView, cho con trỏ nhảy đến vị trí đúng. Tại vị trí này giống như con trỏ đang trỏ đến 1 record trong 1 table, tiếp theo ta cần chỉ rõ ta muốn lấy dữ liệu trên ô thứ mấy của record bằng các tạo idcot chỉ ra cột dữ liệu cần lấy, sau đó dựa vào idcot lấy ra dữ liệu trong cột đó. Dữ liệu đó chính là đường dẫn đến file hình. Đưa nó cho imageView và trả về imageView:

```
public View getView(int position, View view, ViewGroup arg2) {  
    // TODO Auto-generated method stub  
    ImageView imageview;  
    if(view==null)  
    {  
        imageview=new ImageView(context);  
        cursor.moveToPosition(position);  
        int idcot=cursor.getColumnIndex(MediaStore.Images.Thumbnails.DATA);  
        String src=cursor.getString(idcot);  
        imageview.setImageURI(Uri.parse(src));  
  
        imageview.setScaleType(ImageView.ScaleType.FIT_CENTER);  
        imageview.setPadding(8, 8, 8, 8);  
        imageview.setLayoutParams(new GridView.LayoutParams(100, 100));  
    }  
    else  
        imageview=(ImageView)view;  
    return imageview;  
}
```

7. Quay lên onCreate gán adapter cho gridview.

```
gv.setAdapter(new myadapter(this));
```

8. Bắt sự kiện chọn item cho GridView.

```
gv.setOnItemClickListener(new OnItemClickListener() {  
  
    public void onItemClick(AdapterView<?> arg0, View arg1, int position,  
        long arg3) {  
        // TODO Auto-generated method stub  
        cursor.moveToPosition(position);  
        int idcot=cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);  
        String src=cursor.getString(idcot);  
        Toast.makeText(getApplicationContext(),src, Toast.LENGTH_SHORT).show();  
    }  
});
```

9. Chạy thử để test. Khi chọn 1 item chỉ mới hiện ra hộp thông báo bằng Toast.

### Bài 3. Chọn hình trên GridView. (tự làm)

Yêu cầu: phát triển bài 2. Khi chọn vào 1 item trên GridView sẽ chuyển sang 1 activity khác để xem hình lớn, trên activity này có thêm nút “SetWallpaper” khi nhấn nút này sẽ lấy hình hiện tại gán làm wallpaper cho thiết bị.

## Bài 4. Tạo menu để xóa hình trên Gridview (tiếp theo bài 2)

1. Tạo 1 file menu\_layout.xml trong thư mục "res/menu". Layout có 2 mục là xóa và gắn nền, khi người dùng chọn lâu (longClick) 1 đối tượng trên GridView sẽ mở ra menu.

```
*Mediastore2Activity.java  main.xml  layout_menu.xml  mediastore2 Manifest
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/xoa"
        android:title="xoa"
        android:showAsAction="ifRoom"></item>
    <item
        android:id="@+id/gannnen"
        android:title="gannnen"
        />
</menu>
```

2. Override lên 2 hàm là onCreateContextMenu và onContextItemSelected.
3. Hàm onCreateContextMenu viết như sau để chỉ đến file xml làm layout cho menu:

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {
    // TODO Auto-generated method stub
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater menuinflater=getMenuInflater();
    menuinflater.inflate(R.menu.layout_menu, menu);
}
```

4. Trong hàm onCreate đăng kí menu vào GridView như sau:

```
registerForContextMenu(gv);
```

5. Trong class myadapter sửa lại hàm getItem bằng cách di chuyển con trỏ đến vị trí đó, con trỏ sẽ trở đến record, lấy ra cột id sau đó lấy dữ liệu trong cột đó. Hàm trả về src, đây chính là id của hình mà ta chọn trên GridView.

```
public Object getItem(int position) {
    // TODO Auto-generated method stub
    cursor.moveToPosition(position);
    int idcot=cursor.getColumnIndex(MediaStore.Images.Thumbnails._ID);
    String src=cursor.getString(idcot);
    return src;
}
```

6. Trong hàm `onContextItemSelected`, dùng switch đưa ra 2 trường hợp là khi người dùng chọn vào “xoa” hay “gannen” trong menu.

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    AdapterContextMenuInfo info=(AdapterContextMenuInfo)item.getMenuInfo();
    switch(item.getItemId())
    {
        case R.id.xoa:

            break;
        case R.id.gannen:

            break;
    }
    return super.onContextItemSelected(item);
}
```

7. Trong trường hợp chọn vào “xoa” trên menu. Dùng `getItemAtPosition` để lấy ra id của hình được chọn (do hàm `getItem` trong class `myadapter` đã làm ở trên. Tiếp theo tạo ra đối tượng `ContextResolver`. Dùng đối tượng này để gọi hàm `delete`. Hàm `delete` nhận 3 đối số đối số đầu là xóa dữ liệu từ đâu, đối số 2 là mệnh đề `where` (dùng ? để đại diện cho biến), đối số 3 là mảng các biến dùng để thay vào dấu “?” ở đối số đầu. Hiểu đơn giản như sau: “Xóa trong `Context_Uri` những đối tượng nào có `_ID` bằng với `idhinhduocchon`. Tiếp theo là cập nhật lại adapter và `gridview` (biến `md` chính là adapter nhưng phải dùng toàn cục để cập nhật trong hàm này, ta sẽ sửa trên hàm `onCreate` ở bước tiếp).

```
case R.id.xoa:
    String idhinhduocchon=(String)gv.getItemAtPosition(info.position);
    Toast.makeText(getApplicationContext(),idhinhduocchon, Toast.LENGTH_SHORT).show();
    ContentResolver ctr=this.getContentResolver();
    try{
        ctr.delete(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
            MediaStore.Images.Thumbnails._ID+"=?",
            new String[] {idhinhduocchon});

        md.notifyDataSetChanged();
        gv.setAdapter(md);
    }
    catch(Exception e)
    {
        Toast.makeText(getApplicationContext(),e.toString(),Toast.LENGTH_LONG).show();
    }
    break;
```

8. Quay lên trên sửa lại. Tạo 1 biến myadapter toàn cục

```
myadapter md;
```

9. Trong hàm onCreate sửa lại adapter cho gridView như sau:

```
md=new myadapter(this);  
gv.setAdapter(md);
```