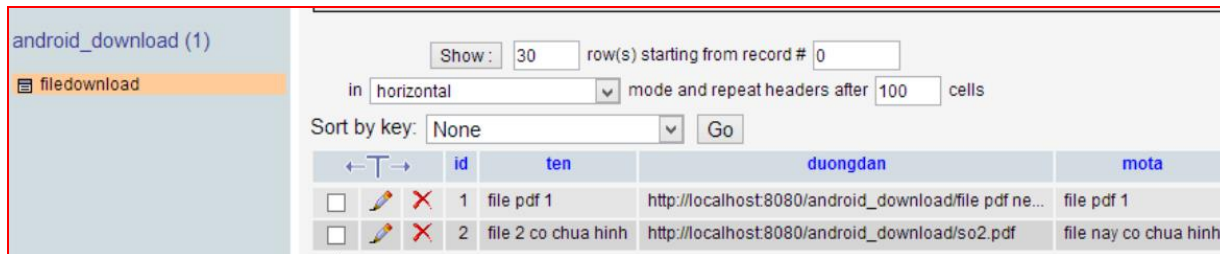


# Download và đọc file PDF được lưu trên webserver (hỗ trợ xem offline)

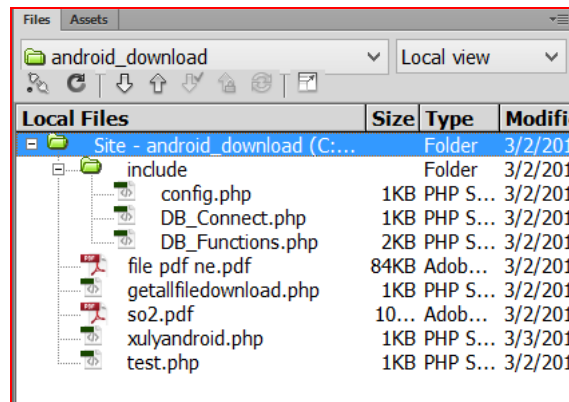
## Trên Webserver

1. Dựa theo bài về login register, xem thêm xóa sửa sản phẩm để tạo database như hình



id	ten	duongdan	mota
1	file pdf 1	http://localhost:8080/android_download/file pdf ne...	file pdf 1
2	file 2 co chua hinh	http://localhost:8080/android_download/so2.pdf	file nay co chua hinh

2. Dựa theo để tạo site tên android\_download trong đó có thư mục include, trong include có config.php giống bài trước, DB\_Connect.php giống bài trước.



Local Files	Size	Type	Modifi
Site - android_download (C:...		Folder	3/2/201
include		Folder	3/2/201
config.php	1KB	PHP S...	3/2/201
DB_Connect.php	1KB	PHP S...	3/2/201
DB_Functions.php	2KB	PHP S...	3/2/201
file pdf ne.pdf	84KB	Adob...	3/2/201
getallfiledownload.php	1KB	PHP S...	3/2/201
so2.pdf	10...	Adob...	3/2/201
xulyandroid.php	1KB	PHP S...	3/3/201
test.php	1KB	PHP S...	3/2/201

3. File DB\_Functions.php đầu tiên viết hàm tên getAllFileDownload để lấy tất cả như sau (câu truy vấn có thể thêm order by để sắp thứ tự).

```
<?
class DB_Functions
{
    private $db;

    function __construct()
    {
        require_once("DB_Connect.php");
        $this->db=new DB_Connect();
        $this->db->connect();
    }
}
```

```
function __destruct()
{
}

//ham lay tat ca cac file download
public function getAllFileDownload()
{
    $sql="select * from filedownload";
    $result=mysql_query($sql);
    return $result;
}
}
?>
```

#### 4. Tạo một file mới tên xulyandroid.php

```
<?
require_once 'include/DB_Functions.php';
$db=new DB_Functions();
$json=array();

if(isset($_POST['tag'])&& $_POST['tag']!='')
{
    $tag=$_POST['tag'];

    $json=array("tag"=>$tag,"thanhcong"=>0,"loi"=>0);

    if($tag=='getallfiledownload')
    {
        getallfiledownload($json,$db);
    }
    else
    {
        echo "yeu cau khong hop le";
    }
}

function getallfiledownload($json,$db)
{
    $result=$db->getAllFileDownload();

    if(mysql_num_rows($result)>0)//co file
    {
        $json["thanhcong"]=1;
        $json["filedownload"]=array(); //mang con

        //duyet tat ca san pham dua vao json
        while($row=mysql_fetch_array($result))
        {
            $filedownload=array();
            $filedownload["id"]=$row["id"];
            $filedownload["ten"]=$row["ten"];
            $filedownload["duongdan"]=$row["duongdan"];
            $filedownload["mota"]=$row["mota"];
```

```
//dua file vao mang
array_push($json["filedownload"],$filedownload);
    }
}
else //khong co file
{
    $json["thanhcong"]=0;
    $json["thongbao"]="khong co file download";
}
echo json_encode($json);
}
?>
```

5. Tạo thêm 1 file tên “test.php” có form để kiểm tra như sau:

```
<form action="xulyandroid.php" method="post">
<input name="tag" type="hidden" value="getallfiledownload" />
<input name="" type="submit" />
</form>
```

6. Chạy file test.php. Nhấn nút để có được chuỗi json



```
{
  "tag": "getallfiledownload",
  "thanhcong": 1,
  "loi": 0,
  "filedownload": [
    {
      "id": "1",
      "ten": "file pdf 1",
      "duongdan": "http://localhost:8080/android_download/file pdf ne.pdf",
      "mota": "file pdf 1"
    },
    {
      "id": "2",
      "ten": "file 2 co chua hinh",
      "duongdan": "http://localhost:8080/android_download/so2.pdf",
      "mota": "file nay co chua hinh"
    }
  ]
}
```

## Thực hiện trên Android

### Tạo file layout

1. Tạo project. Trên file xml chính kéo vào một ListView.
2. Tạo một layout mới tên “list\_item.xml”. Có nội dung là 2 textview. Một TextView chứa id nhưng sẽ không hiện ra, một textview chứa tên của file. **Chú ý:** TextView “fileid” có 1 thuộc tính là android:visibility=”gone” tức là ta có thuộc tính productid nhưng cho nó ẩn đi, nó dùng để gửi dữ liệu id khi ta chọn vào 1 mục trên listview.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/fileid"
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:visibility="gone" />
```

```
<TextView
    android:id="@+id/filename"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Large Text"
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
</LinearLayout>
```

## File JSONParser

File này sẽ có 1 hàm nhận vào một địa chỉ url và kết nối, nhận về chuỗi JSON và sẽ ép chuỗi JSON thành đối tượng JSON

1. Tạo một file mới tên JSONParser.java . File này có hàm getJSONFromUrl nhận vào một url và các đối số, hàm này kết nối lên server gửi yêu cầu POST, nhận về một InputStream (chính là chuỗi JSON) sau đó ép chuỗi JSON về thành JSONObject.

```
public class JSONParser {
    static InputStream is=null;
    static JSONObject jobj=null;
    static String json="";

    public JSONParser()
    {

    }

    //ham nhan vao duong dan va doi so, tra ve doi tuong json
    public JSONObject getJSONFromUrl(String url, List<NameValuePair> cacdoiso)
    {
        try{
            //ket noi goi yeu cau
            DefaultHttpClient httpClient=new DefaultHttpClient();
            HttpPost httpPost=new HttpPost(url);
            httpPost.setEntity(new UrlEncodedFormEntity(cacdoiso));

            //nhan ve du lieu dua vao inputstream
            HttpResponse httpResponse =httpClient.execute(httpPost);
            HttpEntity httpEntity=httpResponse.getEntity();
            is=httpEntity.getContent();

            //doc tu inputstream dua vao doi tuong Json
            BufferedReader reader=new BufferedReader(new InputStreamReader(is,"utf-
8"),8);

            StringBuilder sb=new StringBuilder();
            String line=null;
            while((line=reader.readLine())!=null)
            {
                sb.append(line+"\n");
            }
            is.close();
        }
    }
}
```

```
        json=sb.toString();  
        //Log.d("dulieu", json);  
        jobj =new JSONObject(json);  
  
    }  
    catch (Exception e)  
    {  
  
    }  
    return jobj;  
}  
}
```

### File MyFunctions.java

File này khai báo địa chỉ trên server và chứa các hàm đại diện các chức năng vd ta có hàm getAllFileDownload hàm này sẽ trả về đối tượng JSON chứa danh sách các file có thể download.

1. Tạo một file mới tên MyFunctions.java và viết mã như sau:

```
public class MyFunctions {  
  
    JSONParser jsonparser;  
    String fileurl="http://10.0.2.2:8080/android_download/xulyandroid.php";  
    String getall_tag="getallfiledownload";  
    Context context;  
  
    //ham tao khoi ta doi tuong jsonparser  
    public MyFunctions(Context context)  
    {  
        jsonparser=new JSONParser();  
        this.context=context;  
    }  
  
    //tra ve tat ca du lieu tu bang product  
    public JSONObject getAllFileDownload()  
    {  
        //POST khong can doi so nen ta tao doi so rong  
        List<NameValuePair> cacdoiso=new ArrayList<NameValuePair>();  
        cacdoiso.add(new BasicNameValuePair("tag",getall_tag) );  
  
        JSONObject jobj=jsonparser.getJSONFromUrl(fileurl, cacdoiso);  
        return jobj;  
    }  
}
```

### File MainActivity.java

Quay về file MainActivity, file này sẽ lấy dữ liệu từ server và bỏ vào ListView.

1. Mở file MainActivity khai báo các biến toàn cục

```
ArrayList<FileDownload> ds_file;  
ProgressDialog pg_dialog;  
ListView lv_file;
```

2. Trong hàm onCreate viết thêm như sau:

```
lv_file=(ListView) findViewById(R.id.listView1);  
ds_file=new ArrayList<FileDownload>();
```

3. Xây dựng 2 class nội bên trong class chính để làm adapter, nó sẽ đọc dữ liệu từ ds\_file và đổ vào listview.

```
public static class View_Mot_O  
{  
    TextView tvid;  
    TextView tvname;  
}  
class myadapter extends BaseAdapter{  
    Context context;  
    myadapter(Context c)  
    {  
        context=c;  
    }  
    public int getCount() {  
        return ds_file.size();  
    }  
    public Object getItem(int arg0) {  
        return ds_file.get(arg0);  
    }  
    public long getItemId(int arg0) {  
        return 0;  
    }  
    public View getView(int arg0, View arg1, ViewGroup arg2) {  
        View_Mot_O mot_o;  
        LayoutInflater inf= ((Activity)context).getLayoutInflater();  
        if(arg1==null)  
        {  
            mot_o = new View_Mot_O();  
            arg1 = inf.inflate(R.layout.list_item, null);  
            mot_o.tvid = (TextView) arg1.findViewById(R.id.fileid);  
            mot_o.tvname = (TextView) arg1.findViewById(R.id.filename);  
            arg1.setTag(mot_o);  
        }  
        else  
            mot_o=(View_Mot_O) arg1.getTag();  
  
        mot_o.tvid.setText(ds_file.get(arg0).id);  
        mot_o.tvname.setText(ds_file.get(arg0).ten);  
        return arg1;  
    }  
}
```

4. Xây dựng một class nữa tên xulygetallfile kế thừa từ AsyncTask. Class này sẽ kết nối và lấy dữ liệu JSON về sau đó sẽ đổ dữ liệu vào "ds\_file". Nó còn có thêm ProgressDialog để hiện trước khi kết nối và tắt khi kết nối xong. Sau khi kết nối xong trong onPostExecute nó sẽ gọi setAdapter để đổ dữ liệu vào vào listview. (Chú ý: hàm setAdapter ở đây yêu cầu truyền vào context ta phải truyền vào tên class.this cụ thể là MainActivity.this, không dùng this cũng như getApplicationContext được).

```
class xulygetallfile extends AsyncTask<Void,Void,Void>  
{
```

```
MyFunctions myfunctions;

@Override
protected Void doInBackground(Void... arg0) {
    // TODO Auto-generated method stub
    String thanhcong=null;
    try {
        myfunctions=new MyFunctions(getApplicationContext());
        JSONObject jsonobject=myfunctions.getAllFileDownload();

        thanhcong=jsonobject.getString("thanhcong");

        //doc tat ca du lieu tu json bo vao ArrayList
        if(Integer.parseInt(thanhcong)==1)//thanh cong
        {
            //truy mang ten filedownload trong json
            JSONArray jsonarray=jsonobject.getJSONArray("filedownload");
            //duyet mang
            Log.d("size",jsonarray.length()+"");
            for(int i=0;i<jsonarray.length();i++)
            {
                JSONObject item=jsonarray.getJSONObject(i);
                String id=item.getString("id");
                String ten=item.getString("ten");
                String duongdan=item.getString("duongdan");
                String mota=item.getString("mota");
                FileDownload f=new FileDownload(id,ten,duongdan,mota);
                ds_file.add(f);
            }
        }
        else //that bai
        {
            Toast.makeText(getApplicationContext(),"that bai",
Toast.LENGTH_SHORT).show();
        }
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPreExecute() {
    // TODO Auto-generated method stub
    super.onPreExecute();
    pg_dialog=new ProgressDialog(MainActivity.this);
    pg_dialog.setMessage("dang nap du lieu");
    pg_dialog.setIndeterminate(false);
    pg_dialog.setCancelable(false);//co the cancel bang phim back
    pg_dialog.show();
}

@Override
protected void onPostExecute(Void result) {
    // TODO Auto-generated method stub
    super.onPostExecute(result);
}
```



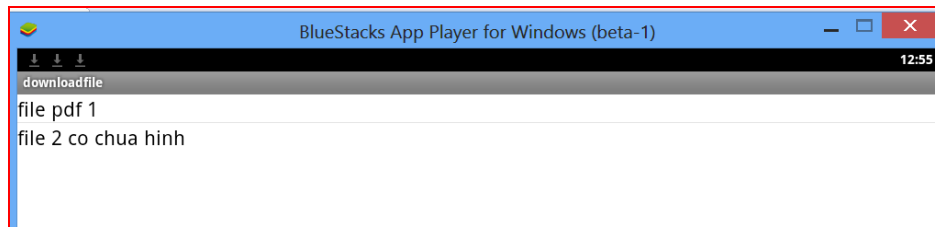
```
pg_dialog.dismiss();

lv_file.setAdapter(new myadapter(MainActivity.this));
}
```

5. Tiếp theo trong onCreate thêm lệnh để kích class “xulygetallfile” chạy

```
new xulygetallfile().execute();
```

6. Chạy chương trình để thấy list view đã show được danh sách các file



Kiểm tra trạng thái kết nối và chuẩn bị đưa vào SQLite để xem offline.

Tiếp theo ta sẽ phát triển để có thể kiểm tra kết nối internet.

1. Mở Manifest và cấp quyền

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
```

2. Tạo một file mới để phục vụ cho SQLite có tên như sau: QuanLyFileDownload.java có mã như bên dưới:

```
public class QuanLyFileDownload extends SQLiteOpenHelper {

    private static final String DB_NAME="quanlyfiledownload";
    private static final String T_NAME="filedownload";
    private static final int DB_VERSION=1;
    public QuanLyFileDownload(Context context, String name,
        CursorFactory factory, int version) {
        super(context, name, factory, version);
        // TODO Auto-generated constructor stub
    }
    public QuanLyFileDownload(Context context)
    {
        super(context, DB_NAME,null, DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        //db.execSQL("drop table if exists " + T_NAME);
        //co the dung lenh create table if not exists
        String sql="create table "+T_NAME+
```



```

        "(" +
            "_id text primary key, " +
            "ten text, " +
            "duongdan text, " +
            "mota text " +
        ")";
        db.execSQL(sql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub
        db.execSQL("drop table if exists "+T_NAME);
        onCreate(db);
    }

    public void taolai()
    {
        SQLiteDatabase db=this.getWritableDatabase();
        db.execSQL("drop table if exists "+T_NAME);
        onCreate(db);
    }

    public void taoDuLieu(String id, String ten, String duongdan, String mota)
    {
        SQLiteDatabase db=this.getWritableDatabase();
        ContentValues noidung=new ContentValues();
        noidung.put("_id", id);
        noidung.put("ten", ten);
        noidung.put("duongdan", duongdan);
        noidung.put("mota", mota);
        String chophepnull=null;
        db.insert(T_NAME, chophepnull, noidung);
    }

    public Cursor getall()
    {
        SQLiteDatabase db=this.getReadableDatabase();
        Cursor c=db.rawQuery("select * from "+T_NAME, null);
        return c;
    }

    public int demsophantu()
    {
        SQLiteDatabase db=this.getReadableDatabase();
        long x= DatabaseUtils.queryNumEntries(db, T_NAME);
        return (int)x;
    }
}

```

3. Mở lại file MainActivity.java và thêm một hàm để kiểm tra kết nối internet như bên dưới, hàm này sẽ trả về true nếu có kết nối internet thông qua wifi hoặc 3G:

```

//ham kiem tra co ket noi internet chua
public boolean checkNetworkStatus() {

    final ConnectivityManager connMgr = (ConnectivityManager)
        this.getSystemService(Context.CONNECTIVITY_SERVICE);
}

```

```
final android.net.NetworkInfo wifi =
connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

final android.net.NetworkInfo mobile =
connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);

if( wifi.isAvailable() )//neu co wifi
    return true;

else if( mobile.isAvailable() ) //neu co 3G
    return true;

else //khong co internet
    return false;
}
```

4. Cũng trong file MainActivity.java, trong hàm onCreate **bỏ lệnh xulygetallfile().execute()** và thêm như bên dưới, phần này dùng để kiểm tra nếu có kết nối internet thì gọi xulygetallfile().execute() còn nếu không có kết nối thì thôi (tạm thời là thế):

```
//neu co internet thi ket noi va tao lai list va dua vao sqlite
//neu khong co internet thi lay truc tiep tu sqlite
if(checkNetworkStatus()==true)//co internet
{
    Toast.makeText(getApplicationContext(),"co net", Toast.LENGTH_LONG).show();
    new xulygetallfile().execute();
}
else if(checkNetworkStatus()==false)//khong co internet
{
    Toast.makeText(getApplicationContext(),"khong net", Toast.LENGTH_LONG).show();
}
}
```

5. Chạy chương trình và kiểm tra nếu có kết nối sẽ lên listview còn nếu không có kết nối internet thì sẽ thông báo bằng Toast (listview trống) đến đây ta vẫn chưa đưa dữ liệu vào SQLite.

### Ghi vào database SQLite (trường hợp có net)

Trong trường hợp có kết nối internet sẽ load dữ liệu từ MySQL về đổ vào ds\_file sau đó làm 2 việc. Thứ 1 từ ds\_file đổ lên listview và việc thứ 2 cần làm là ghi xuống SQLite.

1. Trong MainActivity.java ta thêm 1 hàm mới tên ghivaoSQLite hàm này sẽ phục vụ cho việc ghi dữ liệu từ ds\_file xuống SQLite (phục vụ cho việc offline).







```
public void ghivaoSQLite(Context context)
{
    QuanLyFileDownload ql=new QuanLyFileDownload(context);
    ql.taolai();
    for(int i=0;i<ds_file.size();i++)
    {
        String id=ds_file.get(i).id;
        String ten=ds_file.get(i).ten;
        String duongdan=ds_file.get(i).duongdan;
        String mota=ds_file.get(i).mota;
    }
}
```




```
        ql.taoDuLieu(id, ten, duongdan, mota);
    }
}
```

- Trong hàm onPostExecute ta sẽ gọi hàm trên để sau khi thực thi việc kết nối net xong đổ vào listview đồng thời ghi vào SQLite luôn. Bên dưới lệnh setAdapter ta gọi hàm ghivaoSQLite.

```
lv_file.setAdapter(new myadapter(MainActivity.this));
ghivaoSQLite(getApplicationContext());
```

- Chạy chương trình và xem kết quả, test bằng cách vào DDMS xuất file database ra rồi mở lên xem kết quả, sau đó vào localhost sửa dữ liệu và chạy lại chương trình để xem kết quả, lại xuất file database ra rồi mở lên kiểm tra lại xem kết quả đã cập nhật chưa

←T→	id	ten	duongdan	mota
<input type="checkbox"/>  	1	file pdf 1	http://localhost:8080/android_download/file pdf ne...	file pdf 1
<input type="checkbox"/>  	2	file 2 co chua hinh	http://localhost:8080/android_download/so2.pdf	file nay co chua hinh
<input type="checkbox"/>  	4	ti	http://localhost:8080/android_download/file pdf ne...	ti ne

Check All / Uncheck All With selected:   

rowid	_id	ten	duongdan	mota
1	1	file pdf 1	http://localhost:8080/a...	file pdf 1
2	2	file 2 co chua hinh	http://localhost:8080/a...	file nay co ch
3	4	ti	http://localhost:8080/a...	ti ne

Chú ý giữa database của mysql và database của SQLite. Ta thấy ở MySQL trường id tự tăng và khi ta xóa nó sẽ cấp tiếp theo. Trên SQLite ta thấy có 2 trường id : trường “\_id” là id lấy từ “id” của MySQL, còn trường “rowid” là id trong SQLite tự phát sinh ra, ta thấy mỗi lần làm ta xóa bảng và ghi lại nên rowid luôn tăng và không bị nhảy số. còn \_id thì lấy id của MySQL nên sẽ bị nhảy số. Ta cần phải phân biệt 2 id này.

### Đọc dữ liệu từ SQLite lên (trường hợp không có kết nối)

Nếu không có kết nối internet ta sẽ đọc dữ liệu từ SQLite lên ds\_file sau đó đưa nó vào listview.

- Trong MainActivity.java ta tạo ra một hàm mới tên “doctuSQLite()” hàm này sẽ đọc dữ liệu từ SQLite và đưa vào ds\_file

```
public void doctuSQLite(Context context)
{
    QuanLyFileDownload ql=new QuanLyFileDownload(context);
    Cursor c =ql.getall();
    c.moveToFirst();

    do
    {
        //id của SQLite không lấy được, mà cũng không cần
        String id=c.getString(0);//id của MySQL
        Log.d("dulieu","id"+id);
        String ten=c.getString(1);
        String duongdan=c.getString(2);
```

```
String mota=c.getString(3);
FileDownload f=new FileDownload(id, ten, duongdan, mota);
ds_file.add(f);
}while(c.moveToNext());
}
```

2. Trong hàm onCreate ở phần kiểm tra trạng thái internet trong phần else if (phần không có internet) ta thêm lệnh gọi hàm doctuSQLite() để nạp dữ liệu từ SQLite vào ds\_file sau đó nạp từ ds\_file lên listview

```
else if (checkNetworkStatus() == false) //khong co internet
{
    Toast.makeText(getApplicationContext(), "khong net", Toast.LENGTH_LONG).show();
    doctuSQLite(MainActivity.this);
    lv_file.setAdapter(new myadapter(MainActivity.this));
}
```

3. Trong hàm onCreate ta cũng viết thêm phần bắt sự kiện để khi chọn vào 1 item trên listview sẽ lấy ra được id (id trên MySQL).

```
//xu kien khi click chuot vao mot item tren listview
lv_file.setOnItemClickListener(new OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
        long arg3) {
        //lay textview id(textview nay an)
        //day chinh la id cua MySQL
        TextView tv_idchon=(TextView) arg1.findViewById(R.id.fileid);
        String id=tv_idchon.getText().toString();
        Toast.makeText(getApplicationContext(), "id=" +id ,
        Toast.LENGTH_SHORT).show();
    }
});
```

4. Kiểm tra chương trình. Các vấn đề cần để ý:

#### trên máy ảo

- Việc bật tắt wifi không ổn lắm nên test trên máy thật.
  - Trên máy ảo chuyển qua lại giữa wifi và 3G bằng cách nhấn F8, nếu muốn tắt net thì chuyển sang Airplane.
  - Đôi khi bật wifi hoặc 3G thì nó chỉ nhận được 1 cái có internet . Nên cẩn thận.
- a. Kiểm tra bằng cách bật net và chạy để thấy lấy được dữ liệu từ MySQL -> SQLite.
  - b. Sửa dữ liệu trên MySQL, trên android tắt net và chạy để thấy không có net thì database trên MySQL và SQLite khác nhau.
  - c. Trên android, bật lại net và chạy lại để thấy dữ liệu đã được cập nhật từ MySQL->SQLite.

## Download PDF

1. Trong sự kiện.setOnItemClickListener của lv\_file ta bổ sung thêm phần mã để lấy ra đường dẫn. Phần trên đã lấy được id (trong MySQL), và ta đã lưu nó trong ds\_file do đó dựa vào id ta sẽ truy ra đường dẫn:

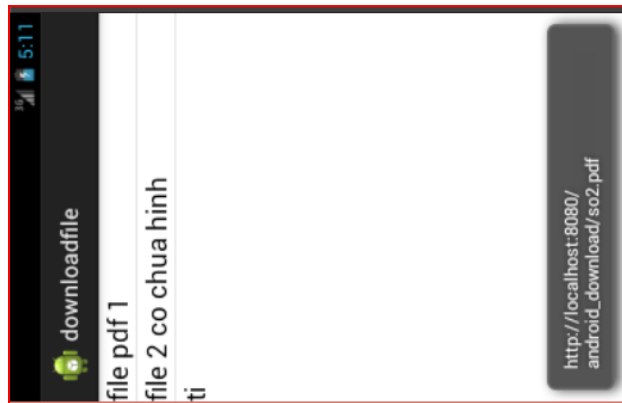
```
String dd_duocchon="";
for(int i=0;i<ds_file.size();i++)
{
    if(ds_file.get(i).id==id)
    {
```

```

        dd_duocchon=ds_file.get(i).duongdan;
        break;
    }
}
Toast.makeText(getApplicationContext(),dd_duocchon, Toast.LENGTH_SHORT).show();

```

2. Chạy thử khi click vào 1 mục trên listview sẽ lấy được id đồng thời lấy được luôn liên kết.



3. Đến đây ta có vấn đề cần sửa. Ta chạy trên máy ảo do đó android sẽ xem ip là 10.0.2.2 chứ không dùng localhost được. Mở MySql lên và toàn bộ đường dẫn chuyển từ localhost sang 10.0.2.2 ngoài ra có port hay không còn tùy theo webserver của bạn. (Một vấn đề nữa là file để download nếu tên có khoảng trắng sẽ bị lỗi, vấn đề này ta sẽ sửa sau).

←T→	id	ten	duongdan	mota
<input type="checkbox"/>	1	file pdf 1	http://10.0.2.2:8080/android_download/file pdf ne....	file pdf 1
<input type="checkbox"/>	2	file 2 co chua hinh	http://10.0.2.2:8080/android_download/so2.pdf	file này co chua hinh
<input type="checkbox"/>	4	ti	http://10.0.2.2:8080/android_download/file pdf ne....	ti ne
<input type="checkbox"/>	5	file txt	http://10.0.2.2:8080/android_download/abc.txt	file txt ne
<input type="checkbox"/>	6	hinh anh	http://10.0.2.2:8080/android_download/a.jpg	hinh anh ne

4. Trong MainActivity.java ta xây 1 hàm mới showPDF, hàm này nhận vào một tên file, nó sẽ truy vào thư mục ExternalStorage vào tiếp thư mục "thumucchuapdf" rồi lấy file ra và mở intent để xem pdf.

(chú ý: máy ảo của eclipse không có chương trình đọc pdf, hoặc bạn kiểm chương trình .apk đọc được pdf rồi dùng command line để cài vào máy ảo (cách này hơi mệt), hoặc ta dùng máy ảo BlueStack rồi lên market và download acrobatreader về cài vào, rồi chạy chương trình trên bluestack).

```

//dua ten file se goi activity de xem pdf
public void showPdf(String tenfile)
{
    File file = new
File(Environment.getExternalStorageDirectory()+"/thumucchuapdf/"+tenfile);
    Log.d("dulieu","..." + file.toString());

    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_VIEW);
    Uri uri = Uri.fromFile(file);
    intent.setDataAndType(uri, "application/pdf");
}

```



```
startActivity(intent);
}
```

5. Khai báo một ProgressDialog toàn cục trong class MainActivity.java

```
ProgressDialog dl_pdialog;
```

6. Trong onCreate khởi tạo ProgressDialog.

```
//tao dialog chophan download file
dl_pdialog= new ProgressDialog(MainActivity.this);
dl_pdialog.setMessage("dang download");
dl_pdialog.setIndeterminate(false);
dl_pdialog.setMax(100);
dl_pdialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
```

7. Trong MainActivity ta xây một class nội kế thừa từ AsyncTask. Trong đó các hàm
- doInBackground** nhận vào mảng chuỗi ta sẽ truyền đường dẫn vào đây. Ta lấy đường dẫn ra (nó là mảng chuỗi nên lấy phần tử đầu), Sau đó từ đường dẫn ta lấy ra tên file. Nếu tên file có khoảng trắng ta thay tất cả khoảng trắng thành %20. Tiếp theo ta truy external storage, tạo một thư mục con tên **"thumucchuapdf"** nếu chưa có. Vào thư mục vừa tạo và tạo một file mới giống tên với tên file trong đường dẫn mà ta đã truyền vào. Sau đó ta dựa vào đường dẫn truyền vào và kết nối đọc dữ liệu theo kiểu Stream từ trên internet đổ vào file trên external storage. Cuối cùng trả về tên file. Trong quá trình đọc ghi ta cũng cập nhật lại progressbar thông qua hàm publicProgress.
  - onPreExecute** ta hiển thị ProgressDialog.
  - onProgressUpdate** ta cập nhật lại ProgressDialog dựa vào giá trị được truyền cho nó.
  - onPostExecute** ta hủy ProgressDialog và gọi hàm showPDF để hiển thị file PDF lên (ở đây ta có thể download tất cả các dạng file từ hình ảnh, video, âm thanh .... Chứ không riêng pdf do đó ta có thể độ lại phần này để hiển thị theo các định dạng khác)

```
class ghiFilePD extends AsyncTask<String,Integer,String>{

    @Override
    protected String doInBackground(String... arg0) {
        // TODO Auto-generated method stub
        String dd=arg0[0];//doi so dau la duong dan
        //cat ra chi lay ten file pdf khong lay http://...
        String tenfile=dd.substring(dd.lastIndexOf("/") +1,dd.length());
        tenfile=tenfile.replace(" ", "%20");

        //lay duong dan external Storage
        String extStorageDirectory =
            Environment.getExternalStorageDirectory().toString();
        File folder = new File(extStorageDirectory, "thumucchuapdf");
        folder.mkdir();//tao thu muc chua

        File file = new File(folder,tenfile);
        try {
            file.createNewFile();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
}
```

```
//////////ket noi va ghi vao file da tao//////////
try {
    //neu duong dan co khoang trang thi thay bang %20
    dd=dd.replace(" ", "%20");
    FileOutputStream f = new FileOutputStream(file);
    URL u = new URL(dd);
    HttpURLConnection c = (HttpURLConnection) u.openConnection();
    c.setRequestMethod("GET");
    c.setDoOutput(true);
    c.connect();
    int dungluong=c.getContentLength();//lay kich thước file
    InputStream in = c.getInputStream();
    byte[] buffer = new byte[1024];
    int da_down=0;//size đã down, đưa ra progressdialog
    int len1 = 0;
    while ((len1 = in.read(buffer)) > 0) {
        da_down+=len1;
        f.write(buffer, 0, len1);
        publishProgress((int) (da_down*100/dungluong));
    }
    f.close();
} catch (Exception e) {
    e.printStackTrace();
    Log.d("dulieu", "luu khong duoc"+ e.toString());
}
//tra tenfile de goi doc file PDF
return tenfile;
}

@Override
protected void onPreExecute() {
    // TODO Auto-generated method stub
    super.onPreExecute();
    dl_pdialog.show();
}

@Override
protected void onProgressUpdate(Integer... values) {
    // TODO Auto-generated method stub
    super.onProgressUpdate(values);
    dl_pdialog.setProgress(values[0]);
}

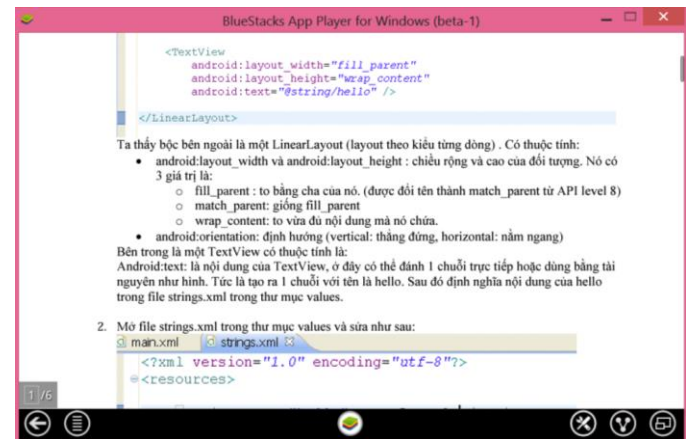
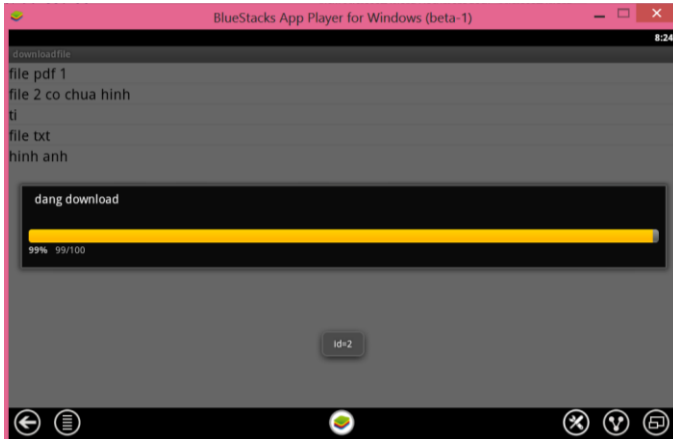
@Override
protected void onPostExecute(String result) {
    // TODO Auto-generated method stub
    super.onPostExecute(result);
    dl_pdialog.dismiss();
    showPdf(result);
}
}
```



8. Cuối cùng trong hàm onCreate, trong sự kiện của lv\_file.setOnItemClickListener thêm lệnh để gọi AsyncTask.

```
ghiFilePD ghifile=new ghiFilePD();  
ghifile.execute(new String[]{dd_duocchon});
```

9. Chạy chương trình trên bluestack để thấy kết quả download có progressdialog, sau khi download xong mở file pdf lên.



10. Có thể phát triển thêm 1 chút nữa trong sự kiện chọn 1 mục trên listview, ta xét có internet hay không hoặc xét xem file đó đã tồn tại trên external storage chưa để quyết định download về rồi mới mở file hay chỉ cần mở file (không cần download).