How to upload file by Multipart Post Entity

Introduction

Uploading large file is risky: your application might crash during uploading or throw StackOverflowException. We provide a solution: use Multipart Entity by Apache to upload these large files and a php script will handle the storage of the file.

However, there is a catch. This is not provided by default in API 8 (SDK 2.2) in your Android library while most of us would be programming our apps on this API.

Requirements

Download the following jar files, we provide the links for your convenience:

HttpMime 4.1:

http://repo1.maven.org/maven2/org/apache/httpcomponents/httpmime/4.1/httpmime-4.1.jar

HttpCore 4.2.1:

http://apache.mirrors.tds.net//httpcomponents/httpcore/binary/httpcomponents-core-4.2.1-bin.tar.gz

HttpClien t4.2.1

http://apache.mirrors.tds.net//httpcomponents/httpclient/binary/httpcomponents-client-4.2.1-bin.tar.gz

Then, add them to your library:

- Create a new folder 'libs' in your Android project.
- Right-click 'libs' and choose Import -> General -> File System, then Next, Browse in the file system to find the library's parent directory (where you downloaded it to).
- Click OK and then click the directory name (not the checkbox) in the left pane, then check the relevant JAR in the right pane. This puts the library into your project.
- Right-click on your project, choose Build Path -> Configure Build Path, then click the Libraries tab, then Add JARs..., navigate to your new JAR in the 'libs' directory and add it.

Advantage and disadvantage

Advantages:

- Uploading large file easily in API 8
- No need to worry about resource lockup during uploading

Disadvantages:

- Require outside libraries

Steps and code snapshot

The magical function:

```java
public void upload() {
    BasicResponseHandler response = new BasicResponseHandler();
    HttpClient client = new DefaultHttpClient();

    // put link to your php script that handles uploading
    HttpPost post = new HttpPost("http://www.YourURL/upload.php");

    // link to your file
    File file = new File(YourFilePath);
    FileBody fBody = new FileBody(file);
    MultipartEntity reqEntity =
                new MultipartEntity(HttpMultipartMode.BROWSER_COMPATIBLE);

    //the variable 'file' here must be the same in server php script
    reqEntity.addPart("file", fBody);
    httpPost.setEntity(reqEntity);

    //receive response from server here
    String result = client.execute(httpPost, response);
    Log.i("File upload", result);
}
```

Server's php script: upload.php (reference in the URL in HttpPost in the above function

```php
<?php
    // variable 'file' must have the same name with the one
    // in java function

    // basic upload procedure in php and return result
    $filename = $_FILES['file']['name'];
    $tmpname=$_FILES['file']['tmp_name'];
    if(move_uploaded_file($tmpname, "upload/$filename")) {
        echo "The file " . basename( $_FILES['file']['name']) .
            " has been uploaded";
    } else {
        echo "There was an error uploading the file " . basename(
$_FILES['file']['name']). ", please try again!";
    }
?>
```