

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

7.6.3 基于TCP协议的Socket通信(2)

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

本节引言：

上节中我们给大家接触了Socket的一些基本概念以及使用方法，然后写了一个小猪简易聊天室的 Demo，相信大家对Socket有了初步的掌握，本节我们来学习下使用Socket来实现大文件的断点续传！这里讲解的是别人写好的一个Socket上传大文件的例子，不要求我们自己可以写出来，需要的时候会用 就好！

1.运行效果图：

1.先把我们编写好的Socket服务端运行起来：

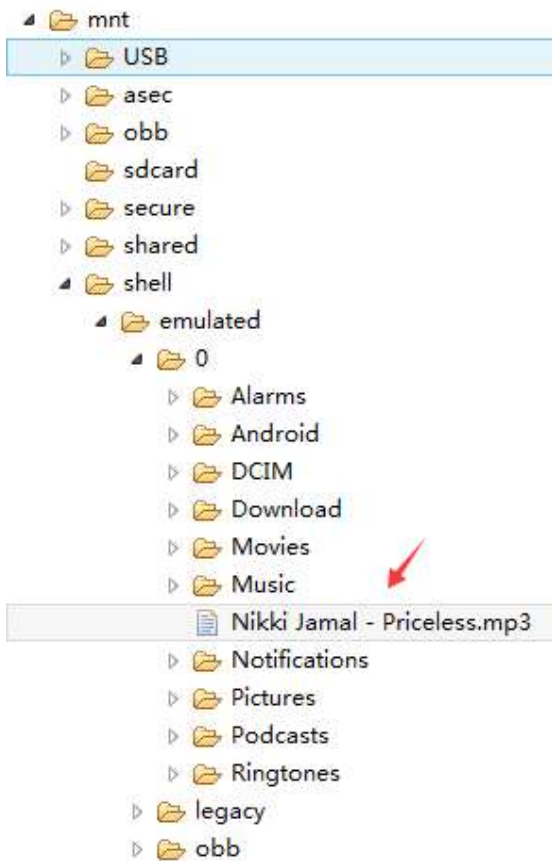


2.将一个音频文件放到SD卡根目录下：

- 1.0 Android基础入门教程
 - 1.0.1 2015年最新Android基...
 - 1.1 背景相关与系统架构分析
 - 1.2 开发环境搭建
 - 1.2.1 使用Eclipse + ADT + S...
 - 1.2.2 使用Android Studio开...
 - 1.3 SDK更新不了问题解决
 - 1.4 Genymotion模拟器安装
 - 1.5.1 Git使用教程之本地仓...
 - 1.5.2 Git之使用GitHub搭建...
 - 1.6 .9(九妹)图片怎么玩
 - 1.7 界面原型设计
 - 1.8 工程相关解析(各种文件...
 - 1.9 Android程序签名打包
 - 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
 - 2.2.1 LinearLayout(线性布局)
 - 2.2.2 RelativeLayout(相对布...
 - 2.2.3 TableLayout(表格布局)
 - 2.2.4 FrameLayout(帧布局)
 - 2.2.5 GridLayout(网格布局)
 - 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5.RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)

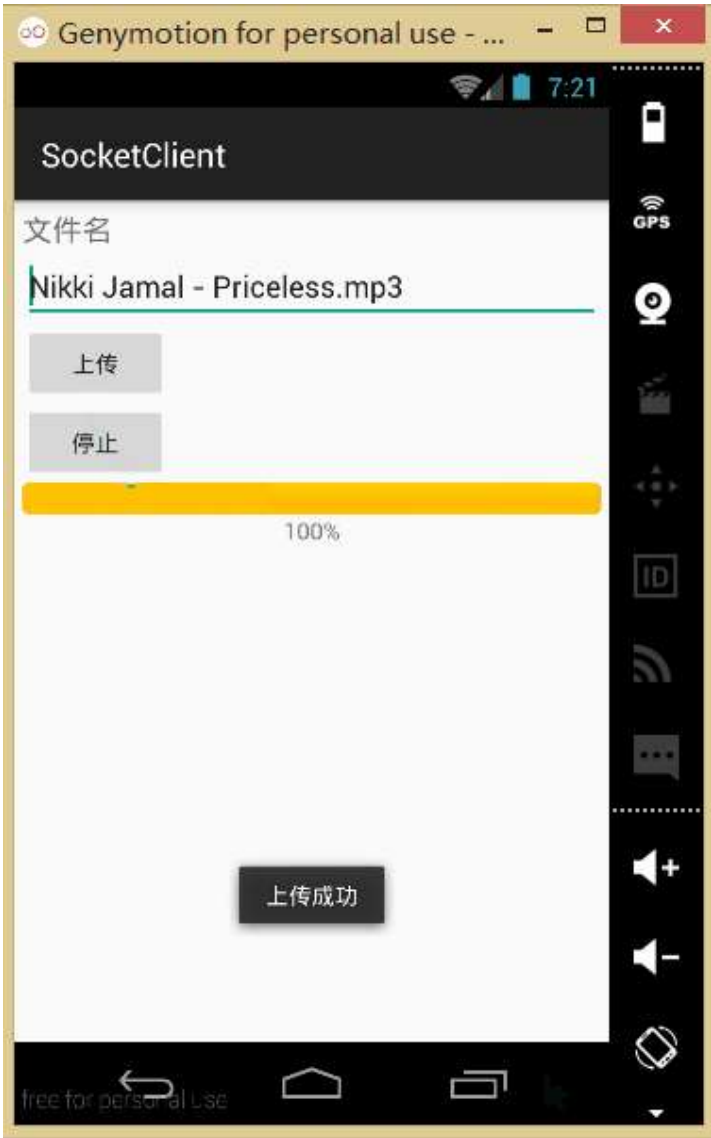


反馈

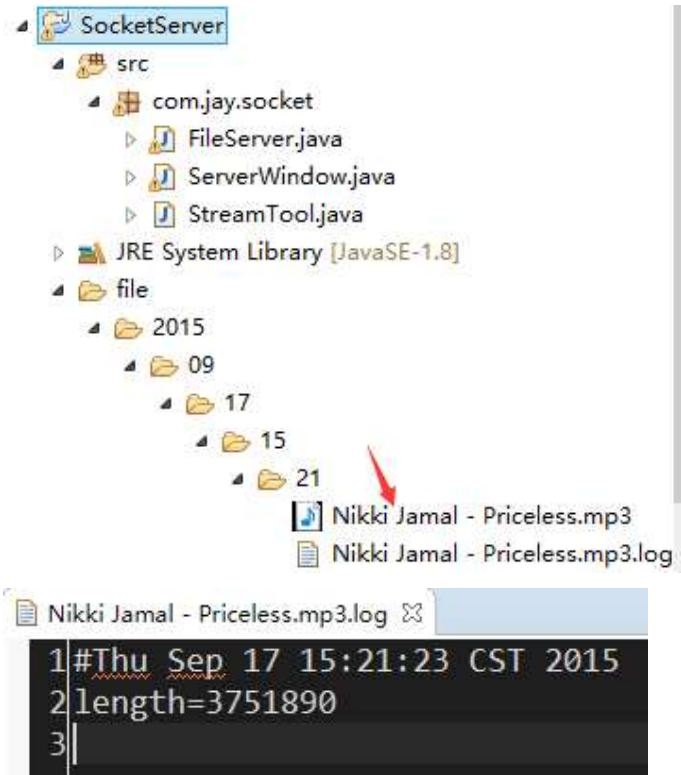


3.运行我们的客户端：

- 2.4.2 Date & Time组件(上)
- 2.4.3 Date & Time组件(下)
- 2.4.4 Adapter基础讲解
- 2.4.5 ListView简单实用
- 2.4.6 BaseAdapter优化
- 2.4.7 ListView的焦点问题
- 2.4.8 ListView之checkbox错...
- 2.4.9 ListView的数据更新问题
- 2.5.0 构建一个可复用的自定...
- 2.5.1 ListView Item多布局的...
- 2.5.2 GridView(网格视图)的...
- 2.5.3 Spinner(列表选项框)...
- 2.5.4 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewPager(翻转视图)...
- 2.5.7 Toast(吐司)的基本使用
- 2.5.8 Notification(状态栏通...
- 2.5.9 AlertDialog(对话框)详解
- 2.6.0 其他几种常用对话框基...
- 2.6.1 PopupWindow(悬浮框...
- 2.6.2 菜单(Menu)
- 2.6.3 ViewPager的简单使用
- 2.6.4 DrawerLayout(官方侧...
- 3.1.1 基于监听的事件处理机制
- 3.2 基于回调的事件处理机制
- 3.3 Handler消息传递机制浅析
- 3.4 TouchListener PK OnTo...
- 3.5 监听EditText的内容变化
- 3.6 响应系统设置的事件(Co...
- 3.7 AsyncTask异步任务
- 3.8 Gestures(手势)
- 4.1.1 Activity初学乍练
- 4.1.2 Activity初窥门径
- 4.1.3 Activity登堂入室
- 4.2.1 Service初涉
- 4.2.2 Service进阶
- 4.2.3 Service精通
- 4.3.1 BroadcastReceiver牛...
- 4.3.2 BroadcastReceiver庖...
- 4.4.1 ContentProvider初探



4.上传成功后可以看到我们的服务端的项目下生成一个file的文件夹，我们可以在这里找到上传的文件：.log那个是我们的日志文件



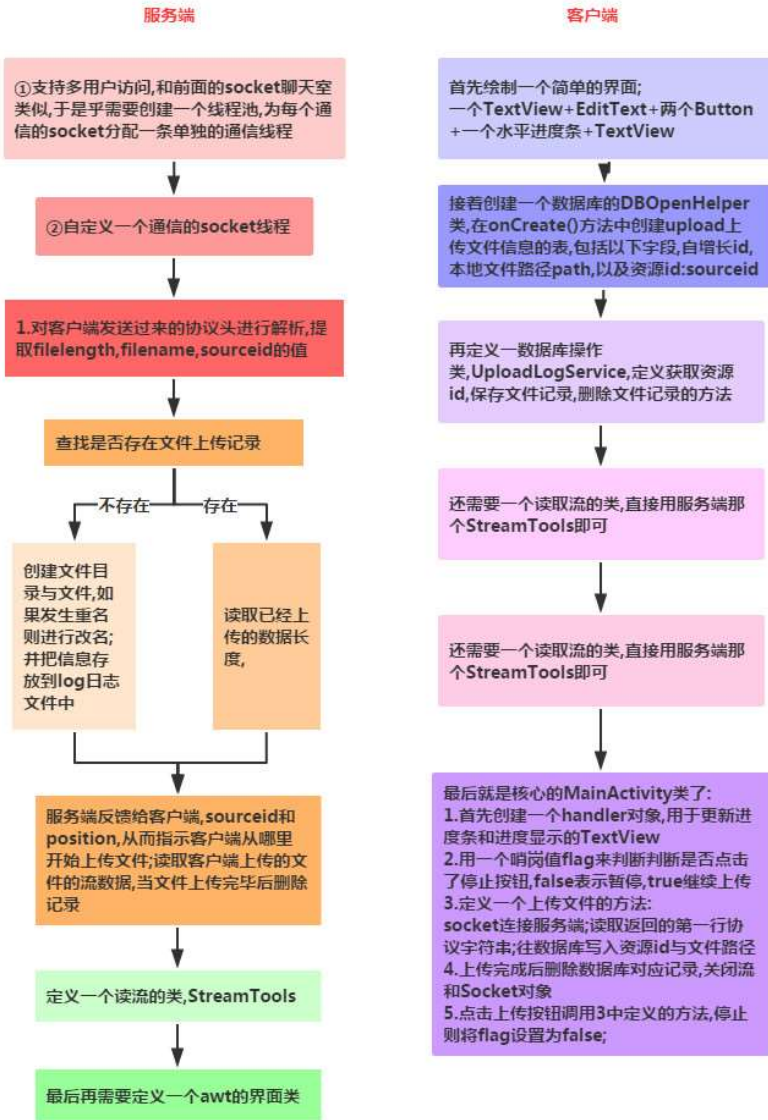
- 4.4.2 ContentProvider再探...
- 4.5.1 Intent的基本使用
- 4.5.2 Intent之复杂数据的传递
- 5.1 Fragment基本概述
- 5.2.1 Fragment实例精讲—...
- 5.2.2 Fragment实例精讲—...
- 5.2.3 Fragment实例精讲—...
- 5.2.4 Fragment实例精讲—...
- 5.2.5 Fragment实例精讲—...
- 6.1 数据存储与访问之——文...
- 6.2 数据存储与访问之——S...
- 6.3.1 数据存储与访问之——...
- 6.3.2 数据存储与访问之——...
- 7.1.1 Android网络编程要学...
- 7.1.2 Android Http请求头与...
- 7.1.3 Android HTTP请求方...
- 7.1.4 Android HTTP请求方...
- 7.2.1 Android XML数据解析
- 7.2.2 Android JSON数据解析
- 7.3.1 Android 文件上传
- 7.3.2 Android 文件下载（1）
- 7.3.3 Android 文件下载（2）
- 7.4 Android 调用WebService
- 7.5.1 WebView(网页视图)基...
- 7.5.2 WebView和JavaScip...
- 7.5.3 Android 4.4后WebVie...
- 7.5.4 WebView文件下载
- 7.5.5 WebView缓存问题
- 7.5.6 WebView处理网页返...
- 7.6.1 Socket学习网络基础准备
- 7.6.2 基于TCP协议的Socket...
- 7.6.3 基于TCP协议的Socket...
- 7.6.4 基于UDP协议的Socke...
- 8.1.1 Android中的13种Draw...
- 8.1.2 Android中的13种Draw...
- 8.1.3 Android中的13种Draw...
- 8.2.1 Bitmap(位图)全解析 P...
- 8.2.2 Bitmap引起的OOM问题
- 8.3.1 三个绘图工具类详解
- 8.3.2 绘图类实战示例

2.实现流程图：

文件断点上传代码流程解析

断点上传的原理:

客户端第一次连接时想服务端发送"Content-length = xx;filename=xx.xx;sourceid="这种格式的字符串,服务端接收后会查找该文件是否有上传记录,如果有的话,返回已经上传的位置,否则返回新生产的sourceid以及position为0,客户端接收返回的字符串后再从指定位置开始上传文件,当然,协议可以由我们自己定义



3.代码示例：

先编写一个服务端和客户端都会用到的流解析类：

StreamTool.java：

```
public class StreamTool {

    public static void save(File file, byte[] data) throws
Exception {
        FileOutputStream outputStream = new FileOutputStream
am(file);

        outputStream.write(data);
    }
}
```

- 8.3.3 Paint API之—— Mask...
- 8.3.4 Paint API之—— Xferm...
- 8.3.5 Paint API之—— Xferm...
- 8.3.6 Paint API之—— Xferm...
- 8.3.7 Paint API之—— Xferm...
- 8.3.8 Paint API之—— Xferm...
- 8.3.9 Paint API之—— Color...
- 8.3.10 Paint API之—— Colo...
- 8.3.11 Paint API之—— Colo...
- 8.3.12 Paint API之—— Path...
- 8.3.13 Paint API之—— Sha...
- 8.3.14 Paint几个枚举/常量值...
- 8.3.15 Paint API之——Type...
- 8.3.16 Canvas API详解(Part 1)
- 8.3.17 Canvas API详解(Part...
- 8.3.18 Canvas API详解(Part...
- 8.4.1 Android动画合集之帧...
- 8.4.2 Android动画合集之补...
- 8.4.3 Android动画合集之属...
- 8.4.4 Android动画合集之属...
- 9.1 使用SoundPool播放音...
- 9.2 MediaPlayer播放音频与...
- 9.3 使用Camera拍照
- 9.4 使用MediaRecord录音
- 10.1 TelephonyManager(电...
- 10.2 SmsManager(短信管理...
- 10.3 AudioManager(音频管...
- 10.4 Vibrator(振动器)
- 10.5 AlarmManager(闹钟服务)
- 10.6 PowerManager(电源服...
- 10.7 WindowManager(窗口...
- 10.8 LayoutInflater(布局服务)
- 10.9 WallpaperManager(壁...
- 10.10 传感器专题(1)——相...
- 10.11 传感器专题(2)——方...
- 10.12 传感器专题(3)——加...
- 10.12 传感器专题(4)——其...
- 10.14 Android GPS初涉
- 11.0 《2015最新Android基...

```

        outputStream.close();
    }

    public static String readLine(PushbackInputStream in) throws IOException {
        char buf[] = new char[128];
        int room = buf.length;
        int offset = 0;
        int c;
        loop: while (true) {
            switch (c = in.read()) {
                case -1:
                case '\n':
                    break loop;
                case '\r':
                    int c2 = in.read();
                    if ((c2 != '\n') && (c2 != -1)) in.unread(c2);
                    break loop;
                default:
                    if (--room < 0) {
                        char[] lineBuffer = buf;
                        buf = new char[offset + 128];
                        room = buf.length - offset - 1;
                        System.arraycopy(lineBuffer, 0, buf, 0, offset);
                    }
                    buf[offset++] = (char) c;
                    break;
            }
        }
        if ((c == -1) && (offset == 0)) return null;
        return String.copyValueOf(buf, 0, offset);
    }

    /**
     * 读取流
     * @param inStream
     * @return 字节数组
     * @throws Exception
     */
    public static byte[] readStream(InputStream inStream) throws Exception {
        ByteArrayOutputStream outStream = new ByteArrayOutputStream();
        byte[] buffer = new byte[1024];
        int len = -1;
        while( (len=inStream.read(buffer)) != -1){
            outStream.write(buffer, 0, len);
        }
    }

```

```

        outStream.close();
        inStream.close();
        return outStream.toByteArray();
    }
}

```

1) 服务端的实现：

socket管理与多线程管理类：

FileServer.java：

```

public class FileServer {

    private ExecutorService executorService;//线程池
    private int port;//监听端口
    private boolean quit = false;//退出
    private ServerSocket server;
    private Map<Long, FileLog> datas = new HashMap<Long, FileLog>(); //存放断点数据

    public FileServer(int port){
        this.port = port;
        //创建线程池，池中具有(cpu个数*50)条线程
        executorService = Executors.newFixedThreadPool(Runtime.getRuntime().availableProcessors() * 50);
    }
    /**
     * 退出
     */
    public void quit(){
        this.quit = true;
        try {
            server.close();
        } catch (IOException e) {
        }
    }
    /**
     * 启动服务
     * @throws Exception
     */
    public void start() throws Exception{
        server = new ServerSocket(port);
        while(!quit){
            try {
                Socket socket = server.accept();
                //为支持多用户并发访问，采用线程池管理每一个用户的连接请求
                executorService.execute(new SocketTask(socket));

            } catch (Exception e) {
                // e.printStackTrace();
            }
        }
    }
}

```

```

private final class SocketTask implements Runnable{
    private Socket socket = null;
    public SocketTask(Socket socket) {
        this.socket = socket;
    }

    public void run() {
        try {
            System.out.println("accepted connection "+ socket.
t.getInetAddress()+ ":"+ socket.getPort());
            PushbackInputStream inStream = new PushbackInput
Stream(socket.getInputStream());
            //得到客户端发来的第一行协议数据: Content-Length=143
253434;filename=xxx.3gp;sourceid=
            //如果用户初次上传文件, sourceid的值为空。
            String head = StreamTool.readLine(inStream);
            System.out.println(head);
            if(head!=null){
                //下面从协议数据中提取各项参数值
                String[] items = head.split(";");
                String filelength = items[0].substring(items
[0].indexOf("=")+1);
                String filename = items[1].substring(items[1
].indexOf("=")+1);
                String sourceid = items[2].substring(items[2
].indexOf("=")+1);
                long id = System.currentTimeMillis();//生产
资源id, 如果需要唯一性, 可以采用UUID
                FileLog log = null;
                if(sourceid!=null && !"".equals(sourceid)){

                    id = Long.valueOf(sourceid);
                    log = find(id);//查找上传的文件是否存在上传
记录

                }
                File file = null;
                int position = 0;
                if(log==null){//如果不存在上传记录, 为文件添加跟
踪记录

                    String path = new SimpleDateFormat("yyyy
/MM/dd/HH/mm").format(new Date());
                    File dir = new File("file/"+ path);
                    if(!dir.exists()) dir.mkdirs();
                    file = new File(dir, filename);
                    if(file.exists()){//如果上传的文件发生重名
, 然后进行改名

                        filename = filename.substring(0, fil
ename.indexOf(".")-1)+ dir.listFiles().length+ filename.substri
ng(filename.indexOf("."));
                        file = new File(dir, filename);
                    }
                    save(id, file);
                }else{// 如果存在上传记录, 读取已经上传的数据长度

```

```

        file = new File(log.getPath()); //从上传记录中得到文件的路径

        if(file.exists()){
            File logFile = new File(file.getParentFile(), file.getName()+".log");
            if(logFile.exists()){
                Properties properties = new Properties();
                properties.load(new FileInputStream(logFile));

                position = Integer.valueOf(properties.getProperty("length")); //读取已经上传的数据长度
            }
        }

        OutputStream outputStream = socket.getOutputStream();

        String response = "sourceid="+ id+ ";position="+ position+ "\r\n";
        //服务器收到客户端的请求信息后，给客户端返回响应信息: sourceid=1274773833264;position=0
        //sourceid由服务器端生成，唯一标识上传的文件，position指示客户端从文件的什么位置开始上传
        outputStream.write(response.getBytes());

        RandomAccessFile fileOutputStream = new RandomAccessFile(file, "rwd");
        if(position==0) fileOutputStream.setLength(Integer.valueOf(filelength)); //设置文件长度
        fileOutputStream.seek(position); //指定从文件的特定位置开始写入数据

        byte[] buffer = new byte[1024];
        int len = -1;
        int length = position;
        while( (len=inStream.read(buffer)) != -1){ //从输入流中读取数据写入到文件中
            fileOutputStream.write(buffer, 0, len);
            length += len;
            Properties properties = new Properties();

            properties.put("length", String.valueOf(length));

            FileOutputStream logFile = new FileOutputStream(new File(file.getParentFile(), file.getName()+".log"));

            properties.store(logFile, null); //实时记录已经接收的文件长度

            logFile.close();
        }
        if(length==fileOutputStream.length()) delete(id);

        fileOutputStream.close();
        inStream.close();
        outputStream.close();

```



```

        file = null;

    }

    } catch (Exception e) {
        e.printStackTrace();
    }finally{
        try {
            if(socket!=null && !socket.isClosed()) socket.close();
        } catch (IOException e) {}
    }
}

public FileLog find(Long sourceid){
    return datas.get(sourceid);
}
//保存上传记录
public void save(Long id, File saveFile){
    //日后可以改成通过数据库存放
    datas.put(id, new FileLog(id, saveFile.getAbsolutePath()));
}
//当文件上传完毕，删除记录
public void delete(long sourceid){
    if(datas.containsKey(sourceid)) datas.remove(sourceid);
}

private class FileLog{
    private Long id;
    private String path;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getPath() {
        return path;
    }
    public void setPath(String path) {
        this.path = path;
    }
    public FileLog(Long id, String path) {
        this.id = id;
        this.path = path;
    }
}
}

```

服务端界面类: **ServerWindow.java** :

```

public class ServerWindow extends Frame {
    private FileServer s = new FileServer(12345);
    private Label label;

    public ServerWindow(String title) {
        super(title);
        label = new Label();
        add(label, BorderLayout.PAGE_START);
        label.setText("服务器已经启动");
        this.addWindowListener(new WindowListener() {
            public void windowOpened(WindowEvent e)
            {
                new Thread(new Runnable() {
                    public void run() {
                        try {
                            s.start
();
                        } catch (Except
ion e) {
                            // e.pr
intStackTrace();
                        }
                    }
                }).start();
            }

            public void windowIconified(WindowEvent
e) {
            }

            public void windowDeiconified(WindowEve
nt e) {
            }

            public void windowDeactivated(WindowEve
nt e) {
            }

            public void windowClosing(WindowEvent e
) {
                s.quit();
                System.exit(0);
            }

            public void windowClosed(WindowEvent e)
            {
            }

            public void windowActivated(WindowEvent
e) {
            }
        });
    }

    /**

```

```

        * @param args
        */
        public static void main(String[] args) throws IOExcepti
on {
            InetAddress address = InetAddress.getLocalHost(
);
            ServerWindow window = new ServerWindow("文件上传
服务端: " + address.getHostAddress());
            window.setSize(400, 300);
            window.setVisible(true);

        }
    }
}

```

2) 客户端(Android端)

首先是布局文件：**activity_main.xml**：

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="5dp">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="文件名"
        android:textSize="18sp" />

    <EditText
        android:id="@+id/edit_fname"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Nikki Jamal - Priceless.mp3" />

    <Button
        android:id="@+id/btn_upload"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="上传" />

    <Button
        android:id="@+id/btn_stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="停止" />

    <ProgressBar
        android:id="@+id/pgbar"
        style="@android:style/Widget.ProgressBar.Horizontal"

```

```
        android:layout_width="fill_parent"
        android:layout_height="40px" />

        <TextView
            android:id="@+id/txt_result"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="center" />
    </LinearLayout>
```

因为断点续传，我们需要保存上传的进度，我们需要用到数据库，这里我们定义一个数据库管理类：**DBOpenHelper.java**：

```
/**
 * Created by Jay on 2015/9/17 0017.
 */
public class DBOpenHelper extends SQLiteOpenHelper {

    public DBOpenHelper(Context context) {
        super(context, "jay.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE IF NOT EXISTS uploadlog (_id integer primary key autoincrement, path varchar(20), sourceid varchar(20))");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }

}
```

然后是数据库操作类：**UploadHelper.java**：

```
/**
 * Created by Jay on 2015/9/17 0017.
 */
public class UploadHelper {
    private DBOpenHelper dbHelper;

    public UploadHelper(Context context) {
        dbHelper = new DBOpenHelper(context);
    }

    public String getBindId(File file) {
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        Cursor cursor = db.rawQuery("select sourceid from uploadlog where path=?", new String[]{file.getAbsolutePath()});
        if (cursor.moveToFirst()) {
```

```

        return cursor.getString(0);
    }
    return null;
}

public void save(String sourceid, File file) {
    SQLiteDatabase db = dbOpenHelper.getWritableDatabase();
    db.execSQL("insert into uploadlog(path,sourceid) values
(?,?)",
        new Object[]{file.getAbsolutePath(), sourceid})
;
}

public void delete(File file) {
    SQLiteDatabase db = dbOpenHelper.getWritableDatabase();
    db.execSQL("delete from uploadlog where path=?", new Ob
ject[]{file.getAbsolutePath()});
}
}

```

对了，别忘了客户端也要贴上那个流解析类哦，最后就是我们的

MainActivity.java了：

```

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private EditText edit_fname;
    private Button btn_upload;
    private Button btn_stop;
    private ProgressBar pgbar;
    private TextView txt_result;

    private UploadHelper upHelper;
    private boolean flag = true;

    private Handler handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            pgbar.setProgress(msg.getData().getInt("length"));
            float num = (float) pgbar.getProgress() / (float) p
gbar.getMax();
            int result = (int) (num * 100);
            txt_result.setText(result + "%");
            if (pgbar.getProgress() == pgbar.getMax()) {
                Toast.makeText(MainActivity.this, "上传成功", To
ast.LENGTH_SHORT).show();
            }
        }
    };

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

        setContentView(R.layout.activity_main);
        bindViews();
        upHelper = new UploadHelper(this);
    }

    private void bindViews() {
        edit_fname = (EditText) findViewById(R.id.edit_fname);
        btn_upload = (Button) findViewById(R.id.btn_upload);
        btn_stop = (Button) findViewById(R.id.btn_stop);
        pgbar = (ProgressBar) findViewById(R.id.pgbar);
        txt_result = (TextView) findViewById(R.id.txt_result);

        btn_upload.setOnClickListener(this);
        btn_stop.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn_upload:
                String filename = edit_fname.getText().toString();

                flag = true;
                if (Environment.getExternalStorageState().equals(
                    Environment.MEDIA_MOUNTED)) {
                    File file = new File(Environment.getExternalStorageDirectory(), filename);
                    if (file.exists()) {
                        pgbar.setMax((int) file.length());
                        uploadFile(file);
                    } else {
                        Toast.makeText(MainActivity.this, "文件并不存在~", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(MainActivity.this, "SD卡不存在或者不可用", Toast.LENGTH_SHORT).show();
                }
                break;
            case R.id.btn_stop:
                flag = false;
                break;
        }
    }

    private void uploadFile(final File file) {
        new Thread(new Runnable() {
            public void run() {
                try {
                    String sourceid = upHelper.getBindId(file);
                    Socket socket = new Socket("172.16.2.54", 12345);

                    OutputStream outputStream = socket.getOutputStream();

                    String head = "Content-Length=" + file.length();

```

```

th() + ";filename=" + file.getName()
        + ";sourceid=" + (sourceid != null
? sourceid : "") + "\r\n";
        outputStream.write(head.getBytes());

        PushbackInputStream inStream = new Pushback
InputStream(socket.getInputStream());
        String response = StreamTool.readLine(inStr
eam);

        String[] items = response.split(";");
        String responseSourceid = items[0].substrin
g(items[0].indexOf("=") + 1);
        String position = items[1].substring(items[
1].indexOf("=") + 1);
        if (sourceid == null) { //如果是第一次上传文件
, 在数据库中不存在该文件所绑定的资源id
                upHelper.save(responseSourceid, file);
            }
            RandomAccessFile fileOutputStream = new Random
AccessFile(file, "r");
            fileOutputStream.seek(Integer.valueOf(position
));

            byte[] buffer = new byte[1024];
            int len = -1;
            int length = Integer.valueOf(position);
            while (flag && (len = fileOutputStream.read(bu
ffer)) != -1) {

                outputStream.write(buffer, 0, len);
                length += len; //累加已经上传的数据长度
                Message msg = new Message();
                msg.getData().putInt("length", length);
                handler.sendMessage(msg);
            }
            if (length == file.length()) upHelper.delet
e(file);

            fileOutputStream.close();
            outputStream.close();
            inStream.close();
            socket.close();
        } catch (Exception e) {
            Toast.makeText(MainActivity.this, "上传异常~
", Toast.LENGTH_SHORT).show();
        }
    }
    }).start();
}
}

```

最后，还有，记得往**AndroidManifest.xml**中写入这些权限哦！

```

<!-- 在SDCard中创建与删除文件权限 -->
<uses-permission android:name="android.permission.MOUNT_UNMOUNT
_FILESYSTEMS"/>

```

```
<!-- 往SDCard写入数据权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<!-- 访问internet权限 -->
<uses-permission android:name="android.permission.INTERNET"/>
```

4.代码下载：

[Socket上传大文件demo](#)

5.本节小结：

本节给大家介绍了基于TCP协议的Socket的另一个实例：使用Socket完成大文件的续传，相信大家对Socket的了解更进一步，嗯，下一节再写一个例子吧，两个处于同一Wifi 下的手机相互传递数据的实例吧！就说这么多，谢谢~



VIPABC 免费请您看电影
体验外教课

立即领取



	<div>在线实例</div> <ul style="list-style-type: none">· HTML 实例· CSS 实例· JavaScript 实例· Ajax 实例· jQuery 实例· XML 实例· Java 实例	<div>字符集&工具</div> <ul style="list-style-type: none">· HTML 字符集设置· HTML ASCII 字符集· HTML ISO-8859-1· HTML 实体符号· HTML 拾色器· JSON 格式化工具	<div>最新更新</div> <ul style="list-style-type: none">· JavaScript 查找...· JavaScript 判断...· 设置 SSH 通过密...· CSS all 属性· Px、Em 换算工具· px,pt,em换算表· px、em、rem 区别...	<div>站点信息</div> <ul style="list-style-type: none">· 意见反馈· 免责声明· 关于我们· 文章归档
				<div>关注微信</div> <div></div> <div>Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1</div>