

[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](#)

7.6.4 基于UDP协议的Socket通信

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

[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\(滚动条\)](#)

本节引言：

本节给大家带来Socket的最后一节：基于UDP协议的Socket通信，在第一节中我们已经详细地比较了两者的区别，TCP和UDP最大的区别在于是否需要客户端与服务端建立连接后才能进行数据传输，如果你学了前两节TCP的，传输前先开服务端，accept，等客户端接入，然后获得客户端socket然后进行IO操作，而UDP则不用，UDP以数据报作为数据的传输载体，在进行传输时首先要把传输的数据定义成数据报(Datagram)，在数据报中指明数据要到达的Socket(主机地址和端口号)，然后再将数据以数据报的形式发送出去，然后就没有然后了，服务端收不收到我就知道了，除非服务端收到后又给我回一段确认的数据报~时间关系就不另外写Android例子了 直接上Java代码~

1.服务端实现步骤：

Step 1：创建DatagramSocket，指定端口号

Step 2：创建DatagramPacket

Step 3：接收客户端发送的数据信息

Step 4：读取数据

示例代码：

```
public class UPDServer {
    public static void main(String[] args) throws IOException {
        /*
         * 接收客户端发送的数据
         */
        // 1.创建服务器端DatagramSocket，指定端口
        DatagramSocket socket = new DatagramSocket(12345);
        // 2.创建数据报，用于接收客户端发送的数据
        byte[] data = new byte[1024]; // 创建字节数组，指定接收的数据包的大小
        DatagramPacket packet = new DatagramPacket(data, data.length);
```



反馈

```

// 3.接收客户端发送的数据
System.out.println("****服务器端已经启动，等待客户端发送数据");

socket.receive(packet); // 此方法在接收到数据报之前会一直阻塞

// 4.读取数据
String info = new String(data, 0, packet.getLength());
System.out.println("我是服务器，客户端说：" + info);

/*
 * 向客户端响应数据
 */
// 1.定义客户端的地址、端口号、数据
InetAddress address = packet.getAddress();
int port = packet.getPort();
byte[] data2 = "欢迎您!".getBytes();
// 2.创建数据报，包含响应的数据信息
DatagramPacket packet2 = new DatagramPacket(data2, data2.length, address, port);
// 3.响应客户端
socket.send(packet2);
// 4.关闭资源
socket.close();
}
}

```

2.客户端实现步骤：

Step 1：定义发送信息

Step 2：创建DatagramPacket，包含将要发送的信息

Step 3：创建DatagramSocket

Step 4：发送数据

```

public class UDPClient {
    public static void main(String[] args) throws IOException {
        /*
         * 向服务器端发送数据
         */
        // 1.定义服务器的地址、端口号、数据
        InetAddress address = InetAddress.getByName("localhost");

        int port = 8800;
        byte[] data = "用户名: admin;密码: 123".getBytes();
        // 2.创建数据报，包含发送的数据信息
        DatagramPacket packet = new DatagramPacket(data, data.length, address, port);
        // 3.创建DatagramSocket对象
        DatagramSocket socket = new DatagramSocket();
        // 4.向服务器端发送数据报
        socket.send(packet);

        /*

```

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

