

UDP协议（用户数据报协议）是无连接的、不可靠的、无序的,速度快。进行数据传输时，首先将要传输的数据定义成数据报（Datagram），大小限制在64k，在数据报中指明数据索要达到的Socket（主机地址和端口号），然后再将数据报发送出去。

DatagramPacket类:表示数据报包

DatagramSocket类: 进行端到端通信的类

## 1、服务器端实现步骤

- ① 创建DatagramSocket，指定端口号
- ② 创建DatagramPacket
- ③ 接受客户端发送的数据信息
- ④ 读取数据

```
/**
 * UDP服务器端
 */
public class UDPServer {
    public static void main(String[] args) throws IOException {
        //1、创建服务器端DatagramSocket，指定端口
        DatagramSocket socket =new DatagramSocket(10010);
        //2、创建数据报，用于接受客户端发送的数据
        byte[] data =new byte[1024];//
        DatagramPacket packet =new DatagramPacket(data,data.length);
        //3、接受客户端发送的数据
        socket.receive(packet);//此方法在接受数据报之前会一致阻塞
        //4、读取数据
        String info =new String(data,0,data.length);
        System.out.println("我是服务器，客户端告诉我"+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

- ① 定义发送信息
- ② 创建DatagramPacket，包含将要发送的信息
- ③ 创建DatagramSocket
- ④ 发送数据

```

/**
 * UDP客户端
 */
public class UDPClient {
    public static void main(String[] args) throws IOException {
        //1、定义服务器的地址、端口号、数据
        InetAddress address =InetAddress.getByName("localhost");
        int port =10010;
        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、向服务器发送数据
        socket.send(packet);

        //接受服务器端响应数据
        //=====
        //1、创建数据报，用于接受服务器端响应数据
        byte[] data2 = new byte[1024];
        DatagramPacket packet2 = new DatagramPacket(data2, data2.length);
        //2、接受服务器响应的数据
        socket.receive(packet2);
    }
}

```

```
String reply = new String(data2, 0, packet2.getLength());  
System.out.println("我是客户端，服务器说: "+reply);  
//4、关闭资源  
socket.close();  
}  
}
```