

UDP网络通信

- ▮ 类 `DatagramSocket` 和 `DatagramPacket` 实现了基于 UDP 协议网络程序。
- ▮ UDP数据报通过数据报套接字 `DatagramSocket` 发送和接收，系统不保证UDP数据报一定能够安全送到目的地，也不能确定什么时候可以抵达。
- ▮ `DatagramPacket` 对象封装了UDP数据报，在数据报中包含了发送端的IP地址和端口号以及接收端的IP地址和端口号。
- ▮ UDP协议中每个数据报都给出了完整的地址信息，因此无须建立发送方和接收方的连接

```
package thread;
```

```
import java.net.DatagramPacket;
```

```
import java.net.DatagramSocket;
```

```
import java.net.InetAddress;
```

```
import org.junit.Test;
```

```
public class udp {
```

```
@Test
```

```
public void test1() {
```

```
    DatagramSocket dSocket = null;
```

```
    DatagramPacket ds = null;
```

```
    try {
```

```
        dSocket = new DatagramSocket();
```

```
        String string = "您好，我是要发送的数据";
```

```
        byte a[] = string.getBytes();
```

```
        //创建一个数据报，每一个数据报不能大于64k，都记录着数据的信息，  
        发送端的端口号
```

```
        //以及要发送到的接收端的IP、端口号
```

```
        ds = new DatagramPacket(a, 0, a.length, InetAddress  
        .getByName("127.0.0.1"), 9090);
```

```
        dSocket.send(ds);
```

```
    } catch (Exception e) {
```

```
        // TODO: handle exception
```

```
        e.printStackTrace();
```

```

    }
    finally {
        if (dSocket != null) {
            dSocket.close();
        }
    }
}

@Test
public void test2() {
    DatagramSocket ds = null;
    DatagramPacket packet = null;
    try {
        ds = new DatagramSocket(9090);
        byte a[] = new byte[1024];
        packet = new DatagramPacket(a, 0, a.length);
        ds.receive(packet);
        String string = new
String(packet.getData(), 0, packet.getLength());
        System.out.println(string);
    } catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    }
    finally {
        if (ds != null) {
            ds.close();
        }
    }
}
}

```