

# 25 网络编程\_Socket通信

JAVAE高级

## 一:Socket概述

- 1、网络上具有唯一标识的IP地址和端口号组合在一起才能构成唯一能识别的标识符套接字
- 2、通信的两端都有Socket，网络通信其实就是Socket间的通信，是两台机器间通信的端点
- 3、数据在两个Socket间通过IO流传输
- 4、Socket在应用程序中创建，通过一种绑定机制与驱动程序建立关系，告诉自己所对应的IP和port
- 5、一般主动发起通信的应用程序属客户端，等待通信请求的为服务端

### 1、Socket类的常用方法

<b>InetAddress getLocalAddress()</b>	<b>返回对方Socket中的IP的InetAddress对象</b>
int getLocalPort()	返回本地Socket中的端口号
InetAddress getInetAddress()	返回对方Socket中IP地址
int getPort()	返回对方Socket中的端口号
void close() throws IOException	关闭Socket，不可在以后的网络连接中使用，除非创建新的套接字
InputStream getInputStream()throws IOException	获取与Socket相关联的字节输入流，用于从Socket中读数据

<b>InetAddress getLocalAddress()</b>	<b>返回对方Socket中的IP的InetAddress对象</b>
OutputStream getOutputStream()throws IOException	获取与Socket相关联的字节输出流，用于向Socket中写数据

## 2、ServerSocket类的常用方法

<b>Socket accept() throws IOException</b>	<b>等待客户端的连接请求，返回与该客户端进行通信的Socket对象</b>
void setSoTimeout(int timeout)throws SocketException	设置accept()方法等待连接的时间为timeout毫秒 若timeout值为0，则表示accept()永远等待 该方法必须在倾听Socket创建后，在accept()之前调用才有效
void close()throws IOException	关闭监听Socket
InetAddress getInetAddress()	返回此服务器套接字的本地地址
int getLocalPort()	返回此套接字在其上监听的端口号
SocketAddress getLocalSocketAddress()	返回此套接字绑定的端点的地址

## 二:UDP传输编程(了解)

### 1、接收端Receive

```

1. // 接收Receive
2. @Test
3. public void getVoid() throws IOException {
4.     DatagramSocket socket = new DatagramSocket(6666); // 创建Socket
指定端口号
5.     DatagramPacket packet = new DatagramPacket(new byte[1024], 1024
); // 创建Packet集装箱
6.
// 指定数组与长度
7.     while (true) {

```

```

8.         socket.receive(packet); // 接收数据
9.         byte[] arr = packet.getData(); // 获取数据
10.        int len = packet.getLength(); // 获取有效的字节个数
11.        String ip = packet.getAddress().getHostAddress(); // 获取ip
    地址
12.        int port = packet.getPort(); // 获取端口号
13.        System.out.println(ip + ":" + port + ":" + new String(arr, 0
    , len));
14.    }
15.    }

```

## 2、发送端Send

```

1.        // 发送Send
2.        @Test
3.        public void getVoid() throws IOException {
4.            Scanner sc = new Scanner(System.in); // 创建录入对象
5.            DatagramSocket socket = new DatagramSocket(); // 创建Socket
6.            while (true) {
7.                String line = sc.nextLine(); // 获取入的字符串
8.                if ("esc".equals(line)) {
9.                    break;
10.                }
11.                DatagramPacket packet = // 创建Packet 集装箱
12.                    new DatagramPacket(line.getBytes(), line.getBytes()
13.                        .length, InetAddress.getByName("127.0.0.1"),
14.                        6666); // 指定数据.长度.地址.端口
15.                socket.send(packet); // 数据发出去
16.            }
17.            socket.close();
18.        }

```