



# Network Programming

---

inter-process communication

Sockets

# Java Sockets Programming



---

The *java.net* Java provides three different types of sockets

- **Socket** class: it is a **Connection-oriented (TCP)**.
- **Datagramsocket** class: it is a **Connectionless (UDP)**.
- **Multicastsocket** class: it is a subclass of the DatagramSocket class and allows data to be sent to multiple recipients.



# Java UDP client/server communication steps

---

- **Step 1** —Create a **datagram socket** object.
- **Step 2** —Create a **buffer** to store the incoming/outgoing datagrams:
- **Step 3** —Create a **datagram packet** object for incoming datagrams:
  
- **Step 4** —Receive/ Send a **datagram**:
- **Step 5** —Get **sender's address** and **port number** from the datagram:
- **Step 6** —Retrieve the data from the buffer:
- **Step 7** — Repeat communication Step-3 to Step-6 if necessary:
- **Step 8**—Close the datagram socket:

```
import java.net.*;
import java.io.*;
import java.util.*;
```

## Client Program 1

```
class UDPClient
```

```
{
```

```
    public static void main(String args[]) throws Exception
```

```
    {
```

```
        DatagramSocket dsoc = new DatagramSocket(1224);
```

```
        InetAddress host = InetAddress.getLocalHost();
```

```
        Date date = new Date();
```

```
        byte b[ ] = date.toString().getBytes();
```

```
        DatagramPacket sp = new DatagramPacket(b, b.length, host, 1224);
```

```
        dsoc.send( sp);
```

```
        dsoc.close();
```

```
    }
```

```
}
```

```
import java.net.*;
import java.io.*;
```

## Server Program 1

```
class UDPServer
{
    public static void main(String args[]) throws Exception
    {
        DatagramSocket dsoc = new DatagramSocket(1224);

        byte buff[ ] = new byte[1024];

        DatagramPacket dpack = new DatagramPacket(buff, buff.length);

        dsoc.receive(dpack);
        System.out.println(new String( dpack.getData() ) );
    }
}
```

## Server Program 2

```
import java.io.*;
import java.net.*;
class UDPServer {
    public static void main(String args[]) throws Exception {
        DatagramSocket ss = new DatagramSocket(9876);
        byte[] rd = new byte[1024];
        byte[] sd;

        while(true) {
            DatagramPacket rp = new DatagramPacket( rd, rd.length );
            ss.receive(rp);
            String sentence = new String( rp.getData(), 0, rp.getLength() );
            InetAddress IPAddress = rp.getAddress();
            int p = rp.getPort();
            String cs = sentence.toUpperCase();
            sd = cs.getBytes();
            DatagramPacket sp=new DatagramPacket(sd,sd.length,IPAddress,p);
            ss.send(sp);
        }
    }
}
```

## Client Program 2

```
import java.io.*;
import java.net.*;

class UDPClient {
    public static void main(String args[]) throws Exception {
        BufferedReader inputfromuser = new BufferedReader(new
        InputStreamReader(System.in));
        DatagramSocket cs = new DatagramSocket();
        InetAddress ips = InetAddress.getByName("hostname");
        byte[] sd;
        byte[] rd = new byte[1024];
        String sentence = inputfromuser.readLine();
        sd = sentence.getBytes();
        DatagramPacket sp = new DatagramPacket(sd, sd.length, ips, 9876);
        cs.send(sp);
        DatagramPacket rp = new DatagramPacket(rd, rd.length);
        cs.receive(rp);
        String ms = new String(rp.getData(),0, rp.getLength());
        System.out.println("FROM SERVER: " + ms);
        cs.close();
    }
}
```

