

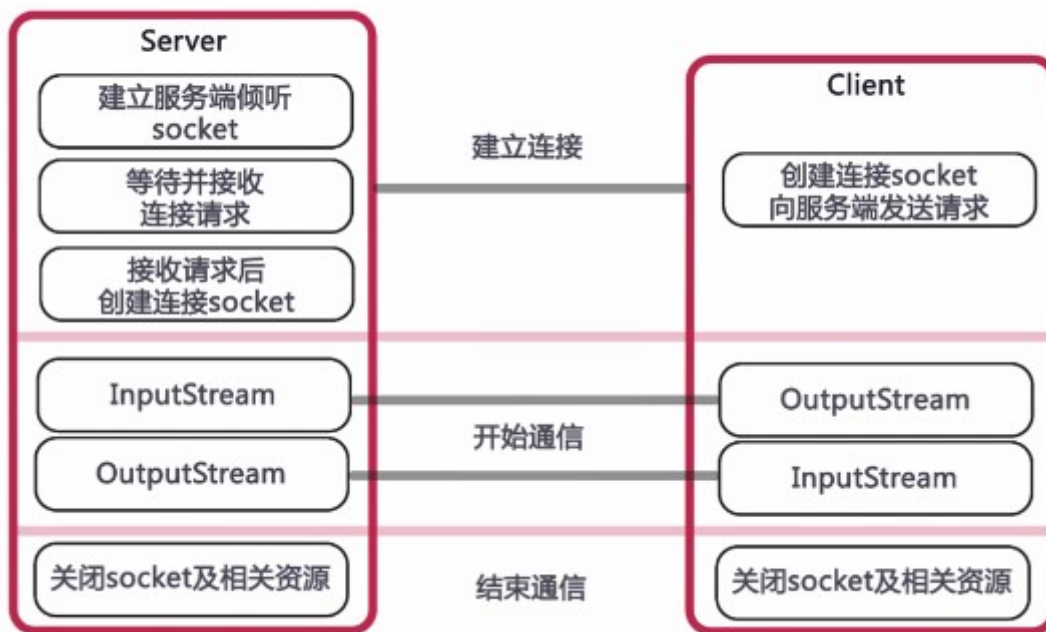
1、TCP协议是面向连接的、可靠的、有序的、以字节流的方式发送数据，通过三次握手方式建立连接，形成传输数据的通道，在连接中进行大量数据的传输，效率会稍低。

2、Java中基于TCP协议实现网络通信的类

客户端的Socket类

服务器端的ServerSocket类

Socket通信模型



3、Socket通信的步骤

- ① 创建ServerSocket和Socket
- ② 打开连接到Socket的输入/输出流
- ③ 按照协议对Socket进行读/写操作
- ④ 关闭输入输出流、关闭Socket

4、服务器端：

- ① 创建ServerSocket对象，绑定监听端口
- ② 通过accept()方法监听客户端请求
- ③ 连接建立后，通过输入流读取客户端发送的请求信息
- ④ 通过输出流向客户端发送乡音信息
- ⑤ 关闭相关资源

/**

* 基于TCP协议的Socket通信，实现用户登录，服务端

*/

```
public class TCPServer {  
    public static void main(String[] args) throws IOException {
```

//1、创建一个服务器端Socket，即ServerSocket，指定绑定的端口，并监听此端

口

//1024-65535的某个端口

```
ServerSocket serverSocket =new ServerSocket(10086);
```

//2、调用accept()方法开始监听，等待客户端的连接

```
Socket socket = serverSocket.accept();
```

//3、获取输入流，并读取客户端信息

```
InputStream is = socket.getInputStream();
```

```
InputStreamReader isr =new InputStreamReader(is);
```

```
BufferedReader br =new BufferedReader(isr);
```

```
String info =null;
```

```
while((info=br.readLine())!=null) {
```

```
    System.out.println("我是服务器，客户端说："+info);
```

```
}
```

```
socket.shutdownInput();//关闭输入流
```

//4、获取输出流，响应客户端的请求

```
OutputStream os = socket.getOutputStream();
```

```
PrintWriter pw = new PrintWriter(os);
```

```
pw.write("欢迎您！");
```

```
pw.flush();
```

//5、关闭资源

```
pw.close();
```

```
os.close();
```

```
br.close();
```

```
isr.close();
```

```
is.close();
```

```
socket.close();
```

```
serverSocket.close();
```

```
}
```

```
}
```

5、客户端：

① 创建Socket对象，指明需要连接的服务器的地址和端口号

② 连接建立后，通过输出流向服务器端发送请求信息

③ 通过输入流获取服务器响应的信息

④ 关闭响应资源

```
/**
```

```
 * TCP客户端
```

```
 */
```

```
public class TCPClient {
```

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

```
//1、创建客户端Socket，指定服务器地址和端口
Socket socket =new Socket("localhost",10086);
//2、获取输出流，向服务器端发送信息
OutputStream os = socket.getOutputStream();//字节输出流
PrintWriter pw =new PrintWriter(os);//将输出流包装成打印流
pw.write("用户名: admin; 密码: 123");
pw.flush();
socket.shutdownOutput();
//3、获取输入流，并读取服务器端的响应信息
InputStream is = socket.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is));
String info = null;
while((info=br.readLine())!=null){
    System.out.println("我是客户端，服务器说: "+info);
}

//4、关闭资源
br.close();
is.close();
pw.close();
os.close();
socket.close();
}
}
```