Java核心API (Socket)

1. Socket相关概念

1）网络通信模型：C/S：client/server，客户端/服务器端；B/S：browser/server，浏览器端/服务器端；C/S结构的优点：应用的针对性强，画面绚丽，应用功能复杂。缺点：不易维护。B/S结构的优点：易于维护。缺点：效果差，交互性不强。

2）Socket：封装着本地的地址，服务端口等信息。ServerSocket：服务端的套接字。

服务器：使用ServerSocket监听指定的端口，端口可以随意指定（由于1024以下的端口通常属于保留端口，在一些操作系统中不可以随意使用，所以建议使用大于1024的端口），等待客户连接请求，客户连接后，会话产生；在完成会话后，关闭连接。

客户端：使用Socket对网络上某一个服务器的某一个端口发出连接请求，一旦连接成功，打开会话；会话完成后，关闭Socket。客户端不需要指定打开的端口，通常临时的、动态的分配一个1024以上的端口。

3）永远都是Socket去主动连接ServerSocket。一个ServerSocket可以接收若干个Socket的连接。网络通信的前提：一定要捕获异常。

4）Socket连接基于TCP/IP协议，是一种长连接（长时间连着）。

5）读取服务器信息会阻塞，写操作不会。

6）建立连接并向服务器发送信息步骤：①通过服务器的地址及端口与服务器连接，而创建Socket时需要以上两个数据。②连接成功后可以通过Socket获取输入流和输出流，使用输入流接收服务端发送过来的信息。③关闭连接。

7）连接服务器：一旦Socket被实例化，那么它就开始通过给定的地址和端口号去尝试与服务器进行连接（自动的）。这里的地址"localhost"是服务器的地址，8088端口是服务器对外的端口。我们自身的端口是系统分配的，我们无需知道。

8）和服务器通信（读写数据）：使用Socket中的getInputStream()获取输入流，使用getOutputStream()获取输出流。

9）ServerSocket构造方法要求我们传入打开的端口号，ServerSocket对象在创建的时候就向操作系统申请打开这个端口。

10）通过调用ServerSocket的accept方法，使服务器端开始等待接收客户端的连接。该方法是一个阻塞方法，监听指定的端口是否有客户端连接。直到有客户端与其连接并接收客户端套接字，否则该方法不会结束。