

Socket

1. java中接收其他通信实体连接请求的是ServerSocket,用于监听Socket连接。使用accept()方法监听, 返回一个**Socket**对象; 构造方法: ServerSocket(端口号)
2. Socket(IP地址,端口号), 构造一个直接连接服务器的Socket, 不能设置超时时间;
3. 使用Socket()无参数的构造器, 之后可以使用connect(一个对象,超时时间), 设置连接超时时间;
4. 使用setSoTimeout(int s)设置读取操作的超时时间, 即连接完成后, 对socket进行通信读写的时间限制; 后面的操作使用**try..catch(SocketTimeoutException e)**来进行限制时间管制。
5. socket提供getInputStream()和getOutputStream()获取输入输出流;
6. 一般在服务端程序代码中, 使用ServerSocket对象的accept()方法获取socket对象后, 再获取输入输出流对客户端进行相应;
7. 在客户端代码中, 使用Socket对象直接连接, 连接后直接获取输入输出流对服务器端进行应答。
8. 加入多线程:

- 服务器端创建一个socket队列(线程同步的list集合), 当有一个相应时生成一个socket并把这个socket放入一个线程任务中, 启动线程, 形成多线程操作;
- 客户端创建一个连接socket对象, 然后传入一个线程任务中实现多线程。
- 记录用户信息的思路: 用一个map记录用户名与socket之间的对应关系; 然后用特殊字符来标记不同的信息。

9. 半关闭socket: shutdownInput()和shutdownOutput()关闭socket中的一个流。

InetAddress类代表IP地址

1. 使用静态方法来获取其实例: getByName(String host);ip地址和getByAddress(byte[] addr);ip地址的二进制数组。
2. 有方法通过实例来获取其ip地址和主机名。

URL对象和URLConnection

1. 使用URLDecoder和URLEncoder来把普通字符串转成MIME字符串。
2. URL对象可以通过:

String getFile(): 资源名; String getHost():主机名; String getPath():路径; int getPost():端口号;
URLConnection.openConnection():获取URLConnection对象; InputStream openInputStream(): 用于下载URL资源。

3. URLConnection进行Get请求和POST请求时; 前者是直接使用connect()方法就可以请求了, 然后通过URLConnection对象获取响应内容; 后者需要使用setDoOutput(true);和setDoInput(true)两个方法; 然后输出流发送请求参数, 输入流获取响应。 urlconnection
https://blog.csdn.net/qq_35716892/article/details/82847211

代理服务器

1. 概念: 在客户端和服务端之间, 客户端访问的实际上是代理服务器, 代理服务器会缓存服务器上的数据, 所以有以下优点: (1) 隐藏真实服务器IP, 突破IP限制; (2) 提高访问速度;
2. 使用Proxy和ProxySelector两个类来实现代理服务器;
3. Proxy使用构造器Proxy(代理服务器类型, 代理服务器地址SocketAddress对象)

- Proxy.Type.DIRECT:直接连接;

- Proxy.Type.HTTP:高级协议代理;
- Proxy.Type.SOCKS:SOCKS代理;

4. 使用: 先创建一个Proxy对象, 然后使用URL对象的openConnection(Proxy)方法获取一个代理的连接; socket的构造器也有类似的。
5. ProxySelector接口是代理选择器, 通过实现该接口的select(url)和connectFailed()方法, 前者是返回一个List对象, 返回代理器集合(如果集合只有一个则为默认的), 后者是连接服务器失败后的方法策略;
6. 还有一种使用代理服务器的方法是设置系统属性: http.proxyHost和http.proxyPort属性。在直接打开连接时, 系统自动调用系统属性设置代理服务器。