黑客派

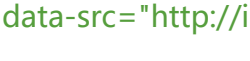
Java 网络通信中的几种方式

作者：JackHoo

原文链接：https://hacpai.com/article/1509881940340

来源网站：黑客派

<h2 id="序言">序言</h2>

最近看过一本 Java 游戏服务器开发的书，以前一直以为游戏是个比较高深的东西，跟 Java Web 开发有很大的区别，但是通读过后发现，有非常多一样的东西，比如网络编程、Java NIO、并发编程设计模式、数据库、缓存、系统运维。这些跟 Web 开发不都是密切相关的吗，可能游戏更注重的游逻辑，视觉效果，还有服务器的 IO 高密集度。
<h3 id="OSI模型">OSI 模型</h3>
说到网络通信就必须知道这个模型了
 
<h3 id="TCP三次握手">TCP 三次握手</h3>

 曾经在知乎上看到有个形象的对话来形容这三次握手
<blockquote>
</blockquote>
「你瞅啥？」
 「瞅你咋地？」
 「来咱俩唠唠。」
然后就唠上了。
<h3 id="TCP和UDP的区别">TCP 和 UDP 的区别</h3>
TCP 特点
<blockquote>
</blockquote>

 面向连接
 安全可靠
 全双工通信
 一对一
 面向流通信

UDP 特点
<blockquote>
</blockquote>

 无连接
 数据无保障
 开销小
 一对一，一对多，多对多都可以(广播)

UDP 可以用在直播协议中
<h3 id="HTTP">HTTP</h3>
HTTP 支持六种请求方法：GET,POST,HEAD,PUT,DELETE,OPTION。在 RESTFul 标准规范中，些方法都会用到。RESTFul 即面向资源、无状态。
<h3 id="Socket">Socket</h3>
Socket 用于描述 IP 地址和端口，是一个通信链的句柄，可以用来实现不同虚拟机或计算机之间通信。连接步骤。
<blockquote>

</blockquote>

 服务器监听
 客户端请求
 连接确认

Java 的 net 包中的 Socket API 可以实现基于 UDP 或 TCP 的 Socket 服务端与客户端。
<h3 id="WebSocket">WebSocket</h3>
WebSocket 随着 HTML5 兴起的一种新协议。实现了浏览器和服务端的全双工通信，服务端可主动向浏览器发送消息。这跟短轮询和长轮询截然不同。当浏览器想要建立 WebSocket 连接时，先起 http 请求，并携带协议升级头信息，服务端接收到后即可建立连接。
 在此有必要介绍下 Stop 协议
<blockquote>
</blockquote>
STOMP 即 Simple (or Streaming) Text Orientated Messaging Protocol，简单(流)文本定向息协议，它提供了一个可互操作的连接格式，允许 STOMP 客户端与任意 STOMP 消息代理（Broke）进行交互。STOMP 协议由于设计简单，易于开发客户端，因此在多种语言和多种平台上得到广泛应用。
 STOMP 协议的前身是 TTMP 协议（一个简单的基于文本的协议），专为消息中间件设。
 STOMP 是一个非常简单和容易实现的协议，其设计灵感源自于 HTTP 的简单性。尽管 STOP 协议在服务器端的实现可能有一定的难度，但客户端的实现却很容易。例如，可以使用 Telnet 登到任何的 STOMP 代理，并与 STOMP 代理进行交互。
 STOMP 协议与 2012 年 10 月 22 日布了最新的 STOMP 1.2 规范。
 要查看 STOMP 1.2 规范，见： https://stomp.github.io/stomp-specification-1.2.html</p

还有一个 SockJS
<blockquote>
</blockquote>
SockJS 是一个 JavaScript 库，提供跨浏览器 JavaScript 的 API，创建了一个低延迟、全双工的览器和 Web 服务器之间通信通道。
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scrpt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
该库有个好处就是，如果当前浏览器不支持 WebSocket 协议，会自动选择回退方案，比如轮询案来保证健壮的通信。
<h3 id="BIO-阻塞式IO">BIO 阻塞式 IO</h3>
<h3 id="NIO-非阻塞式IO">NIO 非阻塞式 IO</h3>
<h3 id="AIO-异步IO">AIO 异步 IO</h3>
<h3 id="Mina和Netty">Mina 和 Netty</h3>