# 什么是WebSocket?

类似于http，websocket是一种网络通信协议，RFC6455定义了它的通信标准。它是html5开始提供的一种在单个TCP连接上进行全双工通讯的协议。

我们平常使用的HTTP协议是一种无状态的，无连接的，单向的应用层协议。它采用的是请求/响应模式，通信请求只能由客户端发起，服务端对请求做出应答。

这样的通讯有一个弊端就是服务器不能主动向客户端发起消息。所以当服务器有连续的状态变化时，客户端想要获取就要通过频繁的调用异步请求轮询。非常浪费资源。

WebSocket连接允许客户端和服务器之间进行全双工通信，以便任一方都可以通过建立的连接将数据推送到另一端。WebSocket 只需要建立一次连接，就可以一直保持连接状态。这相比于轮询方式的不停建立连接显然效率要大大提高。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

可以根据java的socket连接来类比websocket，两边都建立了一直开启的I/O流

基于多线程或多进程的服务器无法适用于 WebSockets，因为它旨在打开连接，尽可能快地处理请求，然后关闭连接。任何实际的 WebSockets 服务器端实现都需要一个异步服务器。

### HTTP 和 WebSocket 有什么关系？

Websocket 其实是一个新协议，跟 HTTP 协议基本没有关系，只是为了兼容现有浏览器的握手规范而已，也就是说它是 HTTP 协议上的一种补充。

Websocket没有cookies

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

在客户端（前端页面）使用WebSocket的时候 不需要使用JavaScript库，实现 WebSockets 的 Web 浏览器将通过 WebSockets 对象公开所有必需的客户端功能（主要指支持 Html5 的浏览器）。

# 如何使用WebSocket？

客户端创建websocket:

@param url 指定的URL protocol 指定了可接受的子协议

var socket = new WebSocket( url, [protocol]) // protocol可选





# 服务器端使用WebSocket

Java后台还是依赖于servlet容器，通过继承tomcat7，Jetty7及以上版本的servlet类。

Spring框架目前也支持websocket的使用。

1.maven项目的pom.xml配置文件引入spring-websocket的jar包。

<propeities>

< spring.version>x.x.xx</ spring.version>

</ propeities>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-websocket</artifactId>

<version>${spring.version}</version>

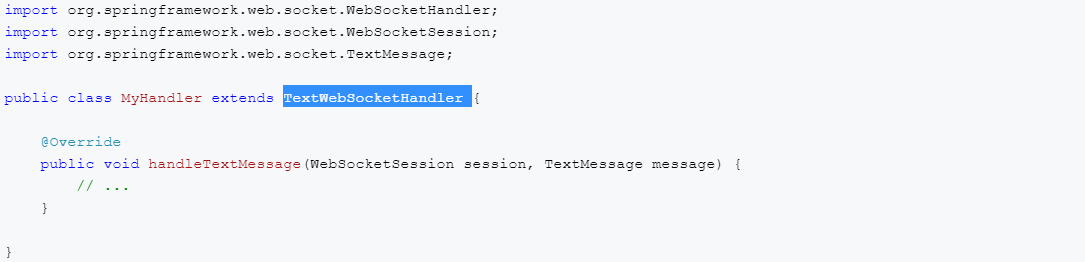
</dependency>

2.配置websocket：可以通过实现WebSocketConfigurer接口注入spring Bean工厂

也可以通过在web.xml文件中配置



通过在对应的路径下使用myHandler方法（实现TextWebSocketHandler接口的方法）完成服务器消息的发送和接受



可以重写方法有如下：

