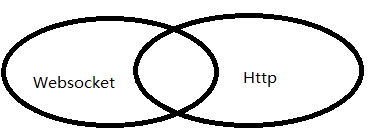
**Websocket原理**

## **1.WebSocket 机制**

以下简要介绍一下 WebSocket 的原理及运行机制。

WebSocket 是 HTML5 一种新的协议。它实现了浏览器与服务器全双工通信，能更好的节省服务器资源和带宽并达到实时通讯，它建立在 TCP 之上，同 HTTP 一样通过 TCP 来传输数据，但是它和 HTTP 最大不同是：

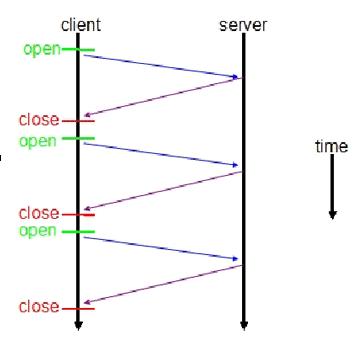
* WebSocket 是一种双向通信协议，在建立连接后，WebSocket 服务器和 Browser/Client Agent 都能主动的向对方发送或接收数据，就像 Socket 一样；
* WebSocket 需要类似 TCP 的客户端和服务器端通过握手连接，连接成功后才能相互通信。



有交集，但是并不是全部。

非 WebSocket 模式传统 HTTP 客户端与服务器的交互如下图所示：

##### **图 1. 传统 HTTP 请求响应客户端服务器交互图**

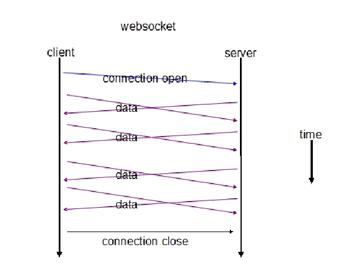


### **ajax轮询**

ajax轮询的原理非常简单，让浏览器隔个几秒就发送一次请求，询问服务器是否有新信息。

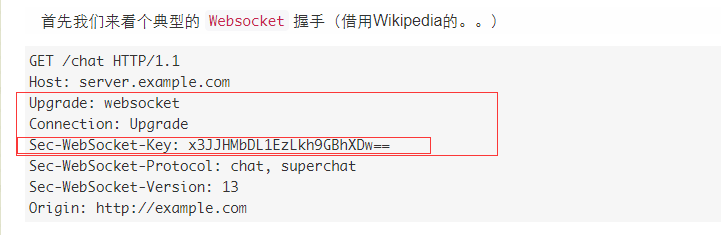
使用 WebSocket 模式客户端与服务器的交互如下图：

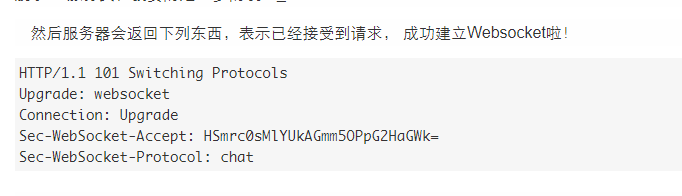
##### **图 2.WebSocket 请求响应客户端服务器交互图**



上图对比可以看出，相对于传统 HTTP 每次请求-应答都需要客户端与服务端建立连接的模式，WebSocket 是类似 Socket 的 TCP 长连接的通讯模式，一旦 WebSocket 连接建立后，后续数据都以帧序列的形式传输。在客户端断开 WebSocket 连接或 Server 端断掉连接前，不需要客户端和服务端重新发起连接请求。在海量并发及客户端与服务器交互负载流量大的情况下，极大的节省了网络带宽资源的消耗，有明显的性能优势，且客户端发送和接受消息是在同一个持久连接上发起，实时性优势明显。

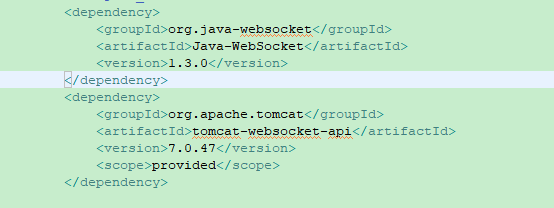
**2.建立连接**

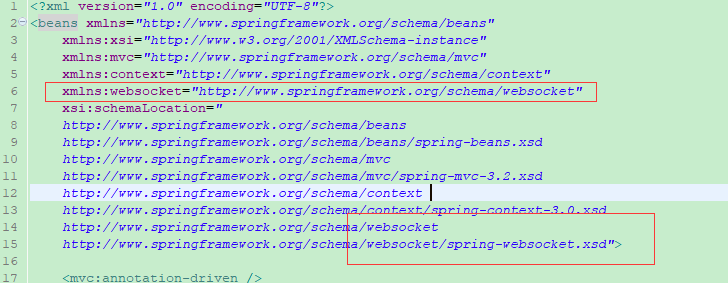
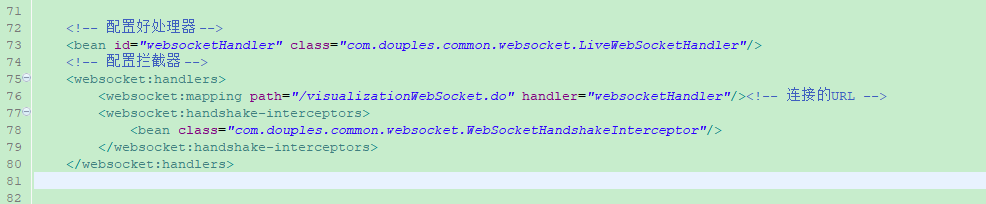
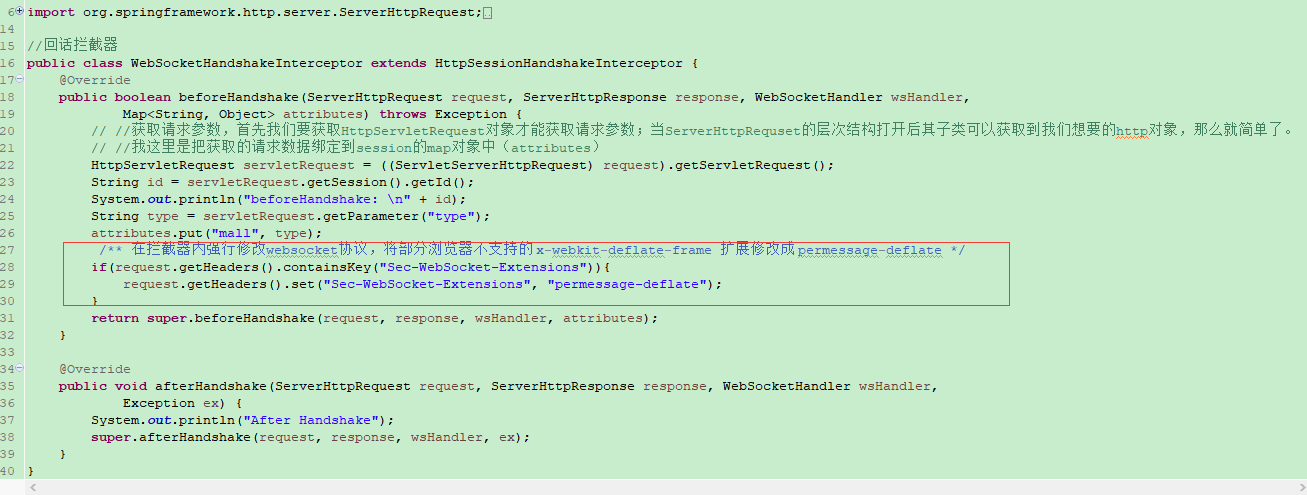




**spring+websocket**

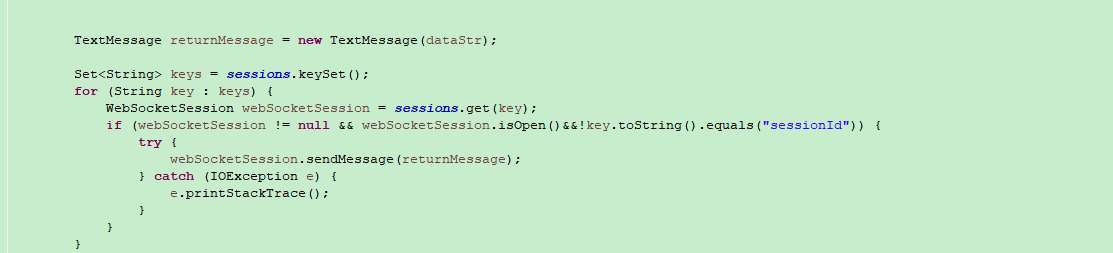
1. 加入 maven依赖 pom.xml



1. 配置spring 文件 加入websocket 
2. 配置拦截器
3. 拦截器处理类

5.创建配置

创建websocket配置相关的类，如我的WebSocketConfig.java，这一个类似于servlet项目中的@ServerEndpoint注解，它的作用就是在html页面中可以直接访问这个类配置的网址，从而进行相关逻辑的处理，而不是经过controller层来管理，有了这个类，WebSocketConfig就相当于是controller层了。

6.发送数据

二 、前端



