# 什么是websocket

Socket：插头，电流从电站引向电器。 服务器向浏览器引数据。 一个道理。

原文：

在WebSocket概念出来之前，如果页面要不停地显示最新的价格，那么必须不停地刷新页面，或者用一段js代码每隔几秒钟发消息询问服务器数据。  
  
 而使用WebSocket技术之后，当服务器有了新的数据，会主动通知浏览器。 如[效果](https://how2j.cn/k/websocket/websocket-tutorial/1625.html#step7056)所示，当服务端有新的比特币价格之后，浏览器立马接收到消息。

原来是浏览器主动刷新或者ajax去请求新数据，而websocket是服务器主动向浏览器发最新的数据

好处：

这种方式会更符合通信逻辑，原来想要一个数据不断更新需要前端向后端建立轮巡，这样效率会很低，10次轮巡9次没有数据，就大大浪费了资源。而且每次都是http请求，附带的“垃圾”请求头会很多。

而websocket这种有新数据了直接后端向前端推的方式，1是websocket不是http方式推数据，带有的垃圾信息很少，2是没有了轮巡，前端的压力会小很多。

1. 节约带宽。 不停地轮询服务端数据这种方式，使用的是http协议，head信息很大，有效数据占比低， 而使用WebSocket方式，头信息很小，有效数据占比高。  
 2. 无浪费。 轮询方式有可能轮询10次，才碰到服务端数据更新，那么前9次都白轮询了，因为没有拿到变化的数据。 而WebSocket是由服务器主动回发，来的都是新数据。  
 3. 实时性，考虑到服务器压力，使用轮询方式不可能很短的时间间隔，否则服务器压力太多，所以轮询时间间隔都比较长，好几秒，设置十几秒。 而WebSocket是由服务器主动推送过来，实时性是最高的

# 配置写法：

原来的Spark项目：

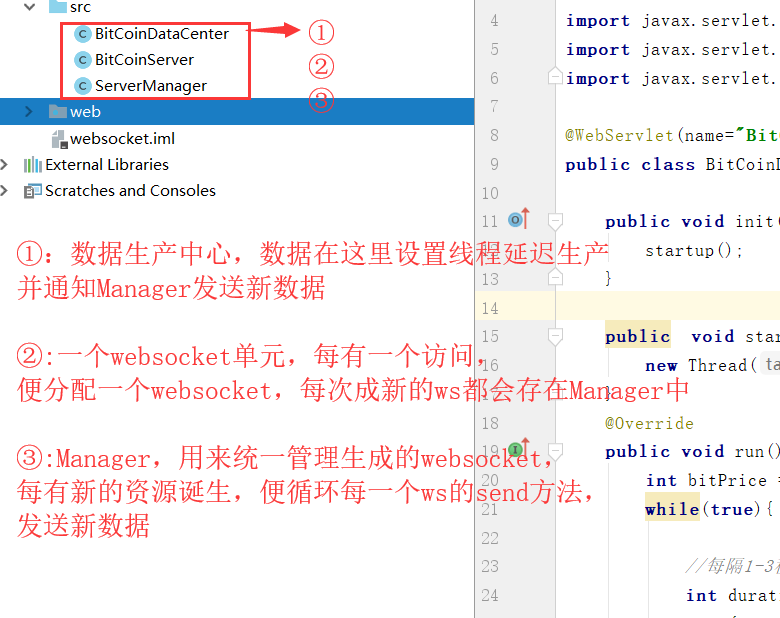






严谨版的应该是这样：

由数据生产中心DataCenter，单元WS，和统一管理WS的Manager组成。（具体写法：<https://how2j.cn/k/websocket/websocket-develop/1628.html#nowhere>）



前端对采用websocket的页面是有配置要求的：



# 注意事项：

如果配置了Struct，ws会被拦截，需要配置一段话：



对于配置了nginx，需要对nginx进行配置，才能正常将ws抛给tomcat

<https://how2j.cn/k/websocket/websocket-nginx/1627.html>

