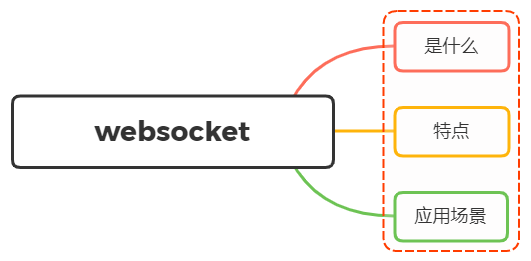
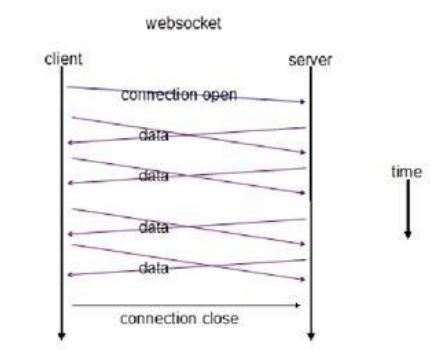
# 面试官：说说对WebSocket的理解？应用场景？



## 一、是什么

WebSocket，是一种网络传输协议，位于OSI模型的应用层。可在单个TCP连接上进行全双工通信，能更好的节省服务器资源和带宽并达到实时通迅

客户端和服务器只需要完成一次握手，两者之间就可以创建持久性的连接，并进行双向数据传输



从上图可见，websocket服务器与客户端通过握手连接，连接成功后，两者都能主动的向对方发送或接受数据

而在websocket出现之前，开发实时web应用的方式为轮询

不停地向服务器发送 HTTP 请求，问有没有数据，有数据的话服务器就用响应报文回应。如果轮询的频率比较高，那么就可以近似地实现“实时通信”的效果

轮询的缺点也很明显，反复发送无效查询请求耗费了大量的带宽和 CPU资源

## 二、特点

### 全双工

通信允许数据在两个方向上同时传输，它在能力上相当于两个单工通信方式的结合

例如指 A→B 的同时 B→A ，是瞬时同步的

### 二进制帧

采用了二进制帧结构，语法、语义与 HTTP 完全不兼容，相比http/2，WebSocket更侧重于“实时通信”，而HTTP/2 更侧重于提高传输效率，所以两者的帧结构也有很大的区别

不像 HTTP/2 那样定义流，也就不存在多路复用、优先级等特性

自身就是全双工，也不需要服务器推送

### 协议名

引入ws和wss分别代表明文和密文的websocket协议，且默认端口使用80或443，几乎与http一致

ws://www.chrono.com  
ws://www.chrono.com:8080/srv  
wss://www.chrono.com:445/im?user\_id=xxx

### 握手

WebSocket也要有一个握手过程，然后才能正式收发数据

客户端发送数据格式如下：

GET /chat HTTP/1.1  
Host: server.example.com  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==  
Origin: http://example.com  
Sec-WebSocket-Protocol: chat, superchat  
Sec-WebSocket-Version: 13

* Connection：必须设置Upgrade，表示客户端希望连接升级
* Upgrade：必须设置Websocket，表示希望升级到Websocket协议
* Sec-WebSocket-Key：客户端发送的一个 base64 编码的密文，用于简单的认证秘钥。要求服务端必须返回一个对应加密的“Sec-WebSocket-Accept应答，否则客户端会抛出错误，并关闭连接
* Sec-WebSocket-Version ：表示支持的Websocket版本

服务端返回的数据格式：

HTTP/1.1 101 Switching Protocols  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=Sec-WebSocket-Protocol: chat

* HTTP/1.1 101 Switching Protocols：表示服务端接受 WebSocket 协议的客户端连接
* Sec-WebSocket-Accep：验证客户端请求报文，同样也是为了防止误连接。具体做法是把请求头里“Sec-WebSocket-Key”的值，加上一个专用的 UUID，再计算摘要

### 优点

* 较少的控制开销：数据包头部协议较小，不同于http每次请求需要携带完整的头部
* 更强的实时性：相对于HTTP请求需要等待客户端发起请求服务端才能响应，延迟明显更少
* 保持创连接状态：创建通信后，可省略状态信息，不同于HTTP每次请求需要携带身份验证
* 更好的二进制支持：定义了二进制帧，更好处理二进制内容
* 支持扩展：用户可以扩展websocket协议、实现部分自定义的子协议
* 更好的压缩效果：Websocket在适当的扩展支持下，可以沿用之前内容的上下文，在传递类似的数据时，可以显著地提高压缩率

## 二、应用场景

基于websocket的事实通信的特点，其存在的应用场景大概有：

* 弹幕
* 媒体聊天
* 协同编辑
* 基于位置的应用
* 体育实况更新
* 股票基金报价实时更新

## 参考文献

* https://zh.wikipedia.org/wiki/WebSocket
* https://www.oschina.net/translate/9-killer-uses-for-websockets
* https://vue3js.cn/interview