

前端的网络通信

1. h5 websocket 移动端

WebSocket是HTML5开始提供的一种浏览器与服务器间进行全双工通讯（full-duplex）的网络技术，可以传输基于信息的文本和二进制的数。

浏览器可以通过JavaScript借助现有的HTTP协议来向服务器发出WebSocket连接请求，当连接建立后，客户端和服务端就可以直接通过TCP连接来直接进行数据交换。这是由于websocket协议本质上就是一个TCP连接，所以在数据传输的稳定性和传输量上有所保证，且相对于以往的轮询和Comet技术在性能方面也有了长足的进步。

有一点需要注意的是虽然websocket在通信时需要借助HTTP，但它本质上和HTTP有着很大的区别：

- WebSocket是一种双向通信协议，在建立连接之后，WebSocket服务端和客户端都能主动向对方发送或者接受数据。
- WebSocket需要先连接，只有再连接后才能进行相互通信。
- 在创建socket实例的时候，new WebSocket()接受两个参数，第一个参数是ws或wss,第二个参数可以选填自定义协议，如果是多协议，可以是数组的方式。
- 在socket.send(data)发送data时，data可以是各种数据，但只有在建立连接后才能使用。
- 在使用socket.close(code,[reason])关闭连接时，code和reason都是选填的。

其中ws表示纯文本通信，而wss表示使用加密信道通信（TCP+TLS）。

ws和wss：

- ws协议：普通请求，占用与HTTP相同的80端口
- wss协议：基于SSL的安全传输，占用与TLS相同的443端口。

2. 后端的 socket

TCP 三次握手

- 第一握：首先客户端发送一个syn，请求连接，
- 第二握：服务器收到之后确认，并发送一个 syn ack应答
- 第三握：客户端接收到服务器发来的应答之后再给服务器发送建立连接的确定。

3. pc socket.io

Socket.IO就是对WebSocket的封装，并且实现了WebSocket的服务端代码。Socket.IO将WebSocket和轮询（Polling）机制以及其它的实时通信方式封装成了通用的接口，并且在服务端实现了这些实时机制的相应代码。也就是说，WebSocket仅仅是Socket.IO实现实时通信的一个子集。Socket.IO简化了WebSocket API，统一了返回传输的API。传输种类包括：

- WebSocket

- Flash Socket
- AJAX long-polling
- AJAX multipart streaming
- IFrame
- JSONP polling.

2. 请描述一下cookies, sessionStorage 和localStorage 的区别?

<https://www.cnblogs.com/cencenyue/p/7604651.html>

共同点:

都是保存在浏览器端, 并且是同源的 (URL的协议、端口、主机名是相同的, 只要有一个不同就属于不同源)

1、cookie数据始终在同源的http请求中携带 (即使不需要), 即cookie在浏览器和服务端间来回传递, 而sessionStorage和localStorage不会自动把数据发送给服务器, 仅在本地保存。cookie数据还有路径 (path) 的概念, 可以限制cookie只属于某个路径下。

2、存储大小限制也不同, cookie数据不能超过4K, 同时因为每次http请求都会携带cookie、所以cookie只适合保存很小的数据, 如会话标识。sessionStorage和localStorage虽然也有存储大小的限制, 但比cookie大得多, 可以达到5M或更大

3、数据有效期不同, sessionStorage仅仅在当前浏览器窗口关闭之前有效; localStorage始终有效, 窗口或者浏览器关闭之后也一直保存, 因此作用持久数据; cookie, 只在设置cookie过期时间之前有效, 即使窗口关闭或者浏览器关闭。

4、作用域不同: sessionStorage在不同的浏览器窗口中不共享, 即使是同一个页面, localStorage在所有的同源窗口中是共享的, cookie也是在所有同源的窗口中共享的。

5、web Storage支持事件通知机制, 可以将数据更新的通知发送给监听者。

6、web Storage的api接口使用更方便。

三者的异同

特性	Cookie	localStorage	sessionStorage
数据的生命期	一般由服务器生成，可设置失效时间。如果在浏览器端生成Cookie，默认是关闭浏览器后失效	除非被清除，否则永久保存	仅在当前会话下有效，关闭页面或浏览器后被清除
存放数据大小	4K左右	一般为5MB	
与服务端通信	每次都会携带在HTTP头中，如果使用cookie保存过多数据会带来性能问题	仅在客户端（即浏览器）中保存，不参与和服务器的通信	
易用性	需要程序员自己封装，源生的Cookie接口不友好	原生接口可以接受，亦可再次封装来对Object和Array有更好的支持	