

# day13 Cookie

## 计算机重要的四个内容

- 数据结构  
数据存储的结构及其逻辑的体现，以及相关数据结构之间的操作（算法）
- 操作系统  
windows（dos命令）、linux（指令操作）
- 计算组成原理  
冯诺伊曼（主板 cpu 内存条 显卡 硬盘）
- 计算机网络  
网络通信就是指代一台计算机到利用传播介质传播到另一台计算机的数据通信过程。

## 计算机网络

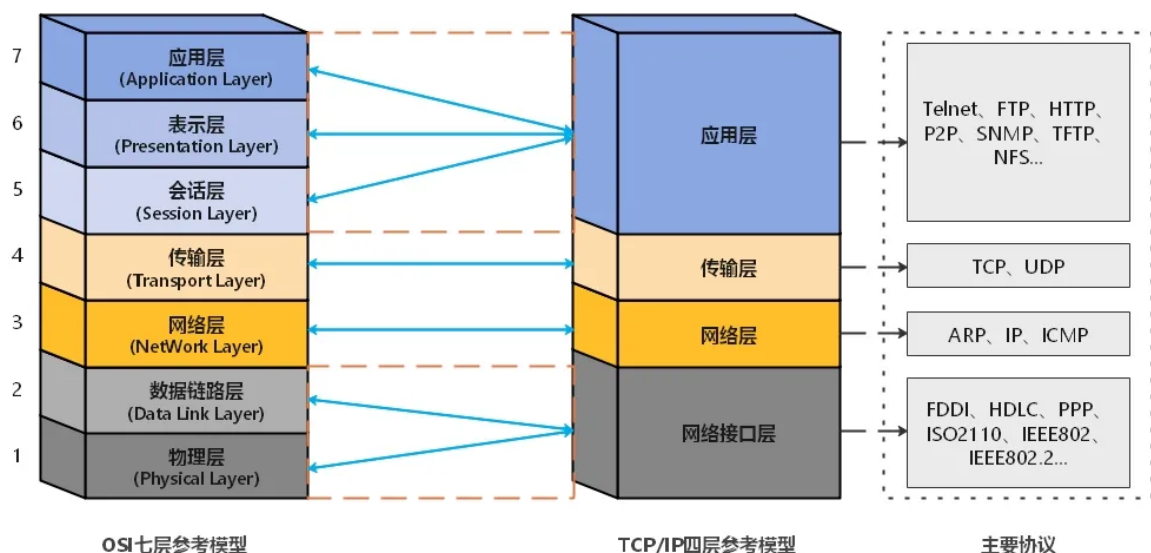
概述：

计算机网络主要概述的是一台一台计算机到利用传播介质传播到另一台计算机的数据传输过程。

主要的内容

- 网络应用 宽带拨号软件
- 传播介质 网线 wifi
- 协议

网络模型图



- 物理层 硬件支持
- 数据链路层 数据传输的接口规范
- 网络层 网络通信  
ip协议 ip4 ip6（DNS分配）
- 传输层 主要协议支持

TCP 一对一传输（必须建立连接）

UDP 丢包的形式（可以一对多 多对多 多对一）

- （应用层 会话层 表示层）应用层相关的内容（应用层相关协议都是来自于对应底层的支持）

## TCP和UDP的区别

TCP 必须建立连接（只能1对1）他是以字节流的形式发送数据的 他的头有64个字节

UDP 不一定要建立连接 通过发送数据报包的形式发送数据 他的头只有8个字节

## 应用层相关的协议

http、https 超文本传输协议

- http 使用明文传输
- https 使用密文传输（ssl进行加密 采用了对称加密及非对称加密 为了安全还提供对应的CA证书）

http有版本差距 http1 http2（http1和http2的区别）

- http1以文件传输形式进行传输（一个请求要有一个连接）
- http2以流的形式进行传输（多路复用 一个连接支持多个请求 同域名下只有一个连接）

## http的讲解

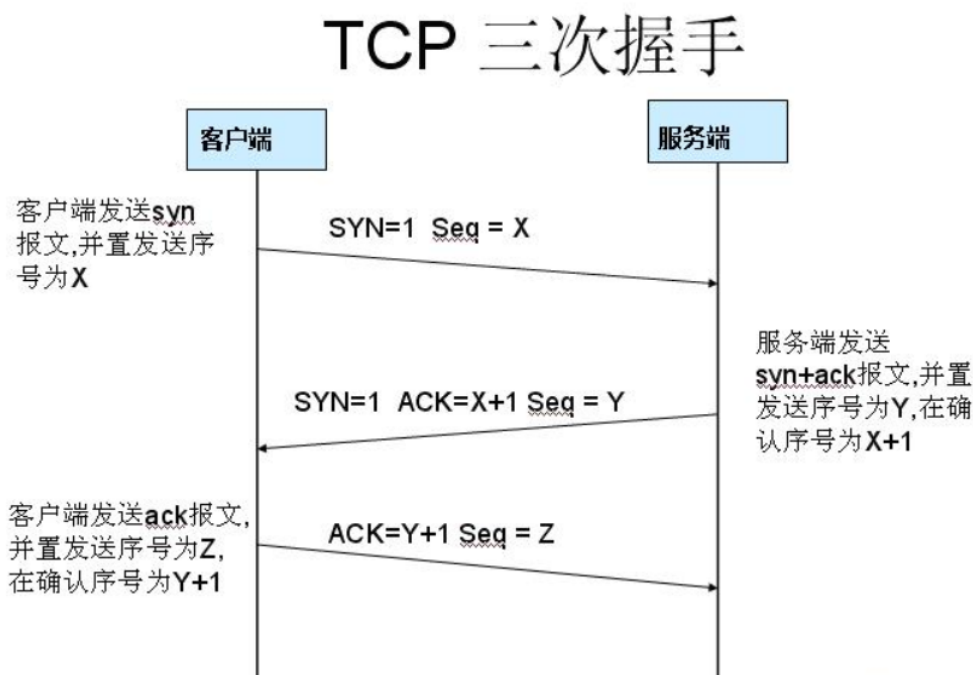
概述：

http 称为超文本传输协议，一般用于网络传输（一般是对应的数据交互），一般交互的数据为JSON格式数据（字符串）、xml（类似于html）。http是基于tcp之上的协议。

## 数据交互的过程（tcp 三次握手 四次挥手）

建立连接的过程 称为三次握手

- 客户端先发送一个信息 告诉他我要建立连接
- 服务端接收到 我已经准备好了
- 客户端收到 那么来建立连接

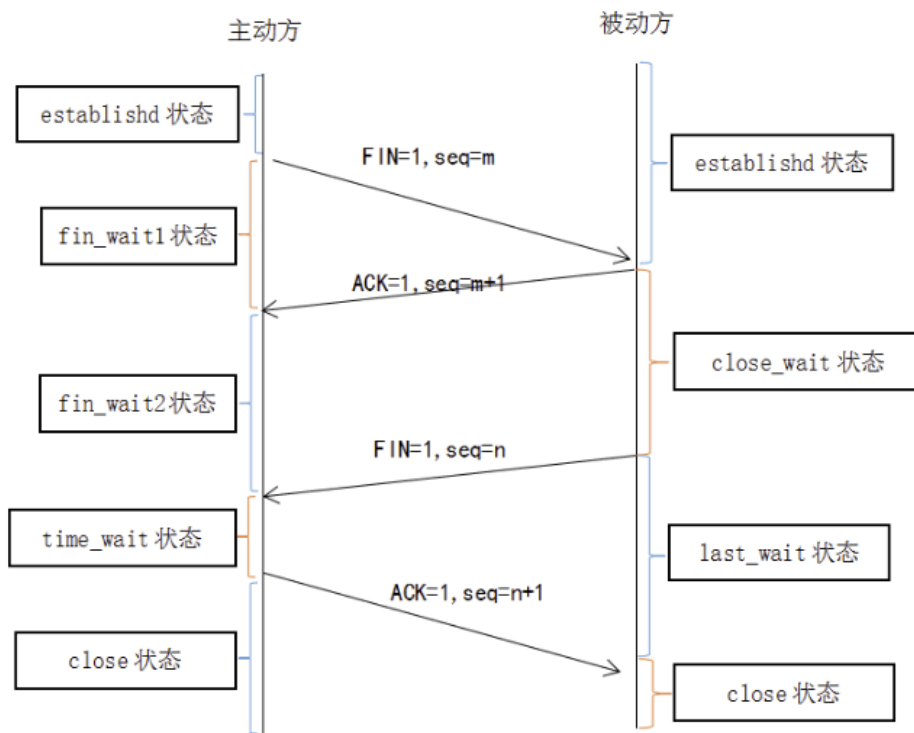


断开连接的过程 称为四次挥手

- 客户端发送断开连接请求

- 服务端接收到断开连接请求
- 服务端断开连接
- 客户端断开连接

## TCP四次挥手



### http的对应的特性

- 无状态 (当前a发送一个请求到b b接收到以后不知道是a 没有记忆能力无法发送请求的人)
- 无连接 (请求发送完建立连接以后会立马断开)
- 长连接 短连接
- ...

### 解决对应的无状态问题

- 主要是通过对应的session (存储在服务端上的) 来解决的, 每个连接的建立都会存在一个对应的 **sessionId** 这个sessionId是由**服务端派发给对应的浏览器的**。连接断开以后对应的sessionId就不在了, 为了解决这个问题他就是sessionId存储在浏览器上, **每次请求都会带上这个 sessionId (cookie)**。**在浏览器用于存储sessionId容器就叫cookie**。如果我对应的服务器关闭那么对应的存储sessionId就没有意义了, 所以cookie也就没有了意义。那么cookie既然存储在对应的浏览器上那么对应的他不应该被销毁, 那么时间一长对应的cookie就会越来越大, 这点来说对我们是不好的。所以为了避免这点他将我们的cookie和session的生命周期做了绑定也就是说**默认情况下对应的cookie的生命周期和session的生命周期是一样的**。session的生命周期是连接一断开他就销毁了。(浏览器一关闭就销毁cookie session)

## cookie讲解

### cookie的结构

cookie里面存储的内容是一个字符串

```
//cookie的名字 name cookie的值 value
//expires 过期时间 如果没有设置默认情况 浏览器关闭就是销毁 如果设置了那么在对应的时间就会销毁
//domain 跨域的地址设置
//path 什么路径下才携带cookie
//secure 是否为https
name=value;expires=Date;domain=地址;path=路径;secure
```

## cookie访问

```
console.log(document.cookie) //显示的过程中只会显示 对应的 name=value
//cookie的添加
document.cookie = `name=张三;expires=${new
Date()};domain=http://127.0.0.1;path=/hello;secure`
```

## cookie的操作方法

```
//添加cookie 修改
//name value 传入一个option 设置对象
function addCookie(name,value,option){
  if(!name){
    throw new Error('参数错误')
  }
  var cookieStr = `${name}=${value}`
  //判断option中是否存在对应的配置
  //instanceof 检索引用数据类型的类型
  if(option){
    if(option.expires instanceof Date){
      cookieStr += `;expires=${option.expires}`
    }
    if(option.domain){
      cookieStr += `;domain=${option.domain}`
    }
    if(option.path){
      cookieStr += `;path=${option.path}`
    }
    if(option.secure){
      cookieStr += `;secure`
    }
  }
  document.cookie = cookieStr
}
//删除
//过期时间设置为当前时间
function removeCookie(name){
  addCookie(name,null,{
    expires:new Date()
  })
}
//封装对应的获取所有的cookie方法
function getCookieValue(name){
  var cookieStr = document.cookie
  var cookie = {}
  //分号分割
  var cookieStrArr = cookieStr.split(';')
  for(var str of cookieStrArr){
    var arr = str.trim().split('=')
```

```
    cookie[arr[0]] = arr[1]
  }
  return cookie[name]
}
```

## 第三方的cookie.js库 (vue2的cookie里引入cookie.js)

- set 设置
- get 获取
- remove 删除
- clear 清除所有
- all 打印所有的cookie

```
cookie.set('hello', '你好')
console.log(cookie.get('hello'))
cookie.remove('hello')
cookie.clear() // 清理所有cookie
cookie.all()
```

<https://toscode.gitee.com/jaywcjlove/cookie.js/>

## cookie的特性

- cookie随请求发送 (每次请求都会携带cookie)
- cookie存储在浏览器上 默认的生命周期为session (浏览器关闭就销毁)
- cookie设置持久化是利用对应的expires属性来设置对应的持久化时间
- cookie的大小只有4k左右 (浏览器不同他就不同)
- cookie里面存在的数据都是字符串
- cookie可以跨域携带 (domain)
- cookie 不安全 (被篡改 被伪造)

## localStorage sessionStorage 本地存储

### 对应的方法

- setItem 设置元素 key : value
- getItem 获取元素 key

```
//只能存字符串
localStorage.setItem('name', 'jack')
console.log(localStorage.getItem('name'))
sessionStorage.setItem('name', 'tom')
console.log(sessionStorage.getItem('name'))
```

### localStorage和sessionStorage的区别

localStorage 关闭浏览器数据依然存在 sessionStorage 浏览器关闭数据就会被清除

### localStorage、sessionStorage 及 cookie的区别

#### 不同点

- cookie会随请求发送 localStorage、sessionStorage 不会
- cookie只有4kb左右 localStorage一般是4-10MB (5MB)
- cookie存储的位置localStorage、sessionStorage存储的位置不一致
- cookie 不能存储复杂的数据 localStorage、sessionStorage跨域存储

- sessionStorage 默认浏览器关闭销毁 cookie默认跟session一样通过expires来设置过期时间  
localStorage永久（手动删除）

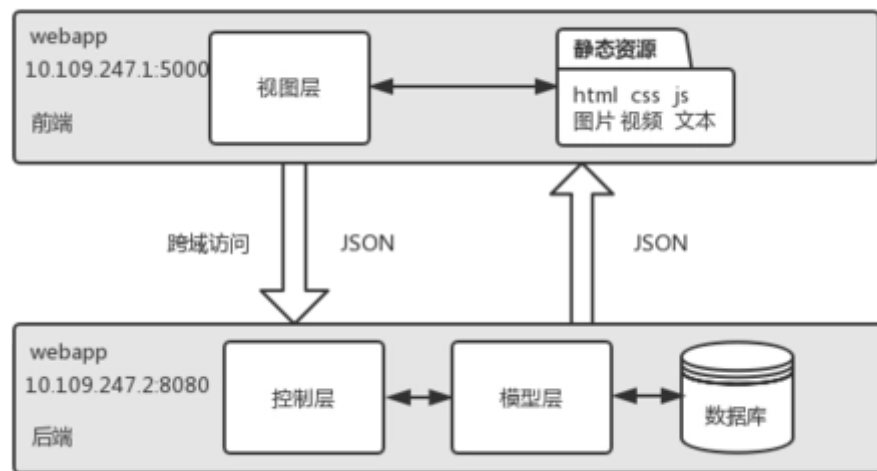
### 相同点

- 都是存在浏览器上的（容易被篡改）
- 都是以字符串的形式进行存储的

## JSON格式

JSON格式是一种数据交互格式，一般后台给我们返回的都是json格式的字符串，但是在js中可以将json格式的字符串变为对象。

### 交互图（主要JSON格式数据进行交互）



- 后端主要提供数据（数据处理（业务）三层模型）
- 前端主要是负责渲染（部分业务前移 三层模型（业务分离））
- 后端给我们返回JSON格式字符串 前端进行解析成对象、然后进行数据渲染

### 主要表现形式 (数组和对象可以多级嵌套)

- 对象 {}
- 数组 []

### 示例

```
var json = {likes:[{name:'苹果',price:50}],age:18}
//获取苹果的价格
console.log(json.likes[0].price)
var jsonArr = [{name:'张三'}, {name:'李四'}, {name:'王五'}]
//获取王五
console.log(jsonArr[2].name)
```

### 练习（将歌词解析处理 然后进行相关的渲染）

```
{"sgc":false,"sfy":false,"qfy":false,"lrc":{"version":11,"lyric":"[00:00.000] 作词：陈思宇/谈晓珍/潘瑛\n[00:00.746] 作曲：Lee Yong Min/Hwang Se Joon\n[00:01.492] Rap词：MC HAN韩勇\n[00:10.985] RAP：\n[00:11.705] Ya boy MC HAN\n[00:14.454] 我弹的钢琴都是为了你弹\n[00:16.577] 弹了那么久还是觉得浪漫\n[00:19.434] 我弹的时候能听到你在唱\n[00:21.412] 感觉上你在这\n[00:22.580] 跟我一起说话\n[00:23.964] 一天到晚 我不停地想\n[00:26.197] You're all that I think of\n[00:27.826] You're all that I want\n[00:28.929] 跟你一起总是让我特别开心\n[00:31.190] 不论发生什么事我永远爱你\n[00:33.716] 如果你突然打了个喷嚏 那一定就是我在想你\n[00:38.058] 如果半夜被手机吵醒 啊那是因为我关心\n[00:42.620] 常常想你说的话是不是别有用心\n[00:47.463] 明明很想相信 却又忍不住怀疑\n[00:52.558] 在你的心里 我是否就是唯一 爱就是有我常烦着你\n[01:01.420] Ho Baby 情话多说一点 想我就多看一眼\n[01:07.028] 表现多一点点 让我能 真的看见\n[01:11.363] Oh Bye 少说一点 想陪你不止一天\n[01:16.430] 多一点 让我 心甘情愿 爱你\n[01:35.835] 喜欢在你的臂弯里胡闹 你的世界是一座城堡\n[01:40.409] 在大头贴画满心号 贴在手机上对你微笑\n[01:45.169] 常常我想说的话你是否听得进去\n[01:50.040] 明明很想生气 却又止不住笑意\n[01:54.734] Oh Oh 在我的心里 你真的就是唯一 爱就是有我常赖着你\n[02:03.785] Ho Baby 情话多说一点 想我就多看一眼\n[02:09.130] 表现多一点点 让我能 真的看见\n[02:14.255] Oh Bye 少说一点 想陪你不止一天\n[02:18.766] 多一点 让我 心甘情愿 爱你\n[02:23.603] 就这样 一天多一点 慢慢地累积感觉\n[02:28.811] 两人的世界 就能够贴近一点\n[02:37.339] Ho Baby 情话多说一点 想我就多看一眼\n[02:42.969] 表现多一点点 让我能 真的看见\n[02:47.728] Oh Bye 少说一点 想陪你不止一天\n[02:52.449] 多一点 让我 心甘情愿 爱你\n[02:57.118] Ho Baby 情话多说一点 想我就多看一眼\n[03:02.191] 表现多一点点 让我能 真的看见\n[03:06.923] Oh Bye 少说一点 想陪你不止一天\n[03:11.429] 多一点 让我 心甘情愿 爱你\n[03:16.584] 多一点 才会慢慢发现 因为你 让我心甘情愿\n[03:26.227] (OT: Nae Yae Gil Eo Bwa) \n"},"tlyric":{"version":0,"lyric":"","code":200}}
```

## 对象转为JSON格式字符串（序列化）

JSON这个类是帮助我们来处理JSON格式数据的 里面提供对应的序列化及反序列化方法

JSON.stringify 将对应的对象转为json格式的字符串

```
//toString 转为的字符串有问题 默认形式的转换字符串用的就是toString  
console.log(obj.toString()) //[object object]  
//利用序列化方法来完成转换成json格式字符串  
var str = JSON.stringify(obj)
```

## JSON格式字符串转为对象（反序列化）

JSON.parse 将对应的字符串转为对象

```
//将对应的字符串转为对象 反序列化方法  
//eval 方法也能转换 执行里面的js （不安全 会直接执行里面的函数）  
// console.log(eval(str))  
//JSON.parse JSON格式字符串必须是双引号修饰的  
console.log(JSON.parse(str)) //传入对应的字符串 返回一个对象
```