

## 课程目标



前端重点



面试重点



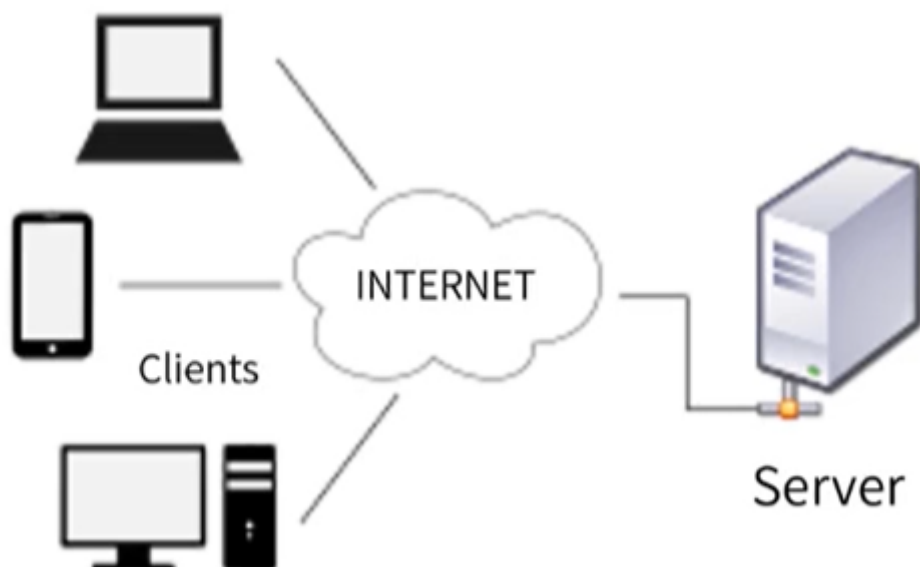
工作日常

## Http协议介绍

### http协议概念

蒂姆伯纳斯李 - 1990年万维网HTTP协议

HTTP协议:超文本传输协议(Hyper Text Transfer Protocol), 超文本协议很好的符合了设计美学



http协议挂载再TCP/IP协议上很容易就实现了, 他一开始就是处理客户端和服务端之间的通信, 但是它提供了一种标准, 中间传文本, 一旦中间能传文本, 整个事件就简化很多了, 因为传二进制太复杂了, 传文本通俗易懂。例如JSON Text 程序都是属于文本协议范围。http请求对应的就是http返回, 这个就没有像TCP/IP那样需要三次握手, 四次挥手这么复杂。早期的主要作用是 网页/JSON/XML/提交表单。

刚开始的设计理念是纯文本+无状态

- 应用层协议
- 纯文本传输

- 无状态(每次请求独立, 请求间互不影响)
- 但是浏览器是需要状态的 所以提供了Cookie Session localStore SessionStore 等)

## HTTP历史

■ 1991 HTTP 0.9



■ 1996 HTTP 1.0

■ 1999 HTTP 1.1

■ 2015 HTTP 2.0

## 设计Http协议的基础因素

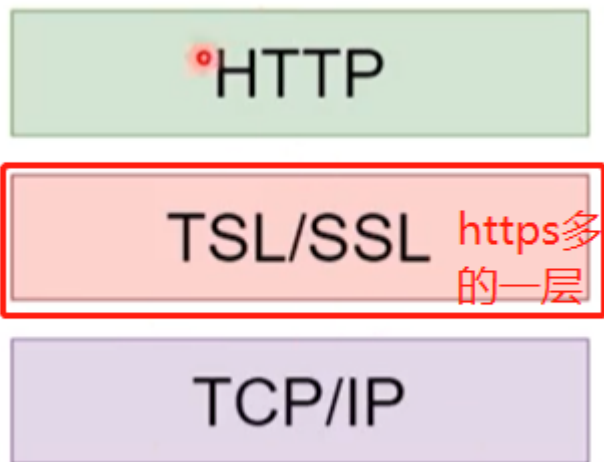
1. 宽带(基础网络-线路 设备等)
2. 延迟(浏览器 DNS查询 建立连接-TCP三次握手) 前端重要的两个指标: 一个是跳出率(用户打开网页, 加载慢的话, 跑了....) 另外一个用户的可交互时间

## 设计Http协议考虑因素

1. 缓存 (http1.0 提供缓存机制 如 If-Modified-Since http1.1提供E-Tag等高级缓存策略)
2. 宽带优化(http1.1利用range头获取文件的某个部分 http1.1利用长连接让多个请求在一个TCP连接上排队 http2.0 利用多路复用技术同时传输多个请求)
3. 压缩(http2.0 主流web服务器, 如nginx/express等提供gzip压缩功能 http2.0采用二进制传输 头部使用HPACK算法压缩)
4. 安全性HTTPS(在HTTP和TCP/IP之间增加了TSL/SSL层 数据传输加密(非对称+对称加密))

## HTTPS

HTTPS安全超文本传输协议(Hyper Text Transfer Protocol Secure)



本来的http协议应用层 就建立在 TCP/IP(传输网络层)上, https就在HTTP和TCP/IP中间多加了一层TSL/SSL(SSL安全套接层 TSL传输层安全协议 需要的在客户端安装证书)主要的作用就是数据加密传输, 防止各种手段攻击(信息泄露,篡改)

## Node.js实战http请求

### header和Body(实战)

- HTTP协议是一个文本传输协议, 传输的内容是人类可读的文本, 大体文本分成两部分
  - 请求头(Header)/返回头
  - 消息体(Body)
- 观察node实现http的基础协议

```
1 //实现一个http协议
2 const net =require('net')
3 const response =
4   `HTTP/1.1 200 OK
5   Data: Tue, 30 Jun 2021 01:00:00 GMT
6   Content-Type: text/plain
7   Connection: Closed
8
9   Hello world
10 `
11
12
13 const server = net.createServer((socket)=>{
14   socket.end(response)
15 })
```

```
16 server.listen(80,()=>{
17   console.log(1212)
18 })
```

## chrome/curl/postman/whistle工具初探

Google开发的免费浏览器

Google开发者工具具有强大的调试能力

cUrl工具 传输一个URL(和服务器交互的工具) 支持多种协议(HTTP HTTPS FTP .....

```
C:\Users\yangke\Desktop\test>curl http://www.baidu.com
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html; charset=utf-8><meta http-equiv=
href=http://s1.bdstatic.com/r/www/cache/bdorz/baidu.min.css><title>百度一下, 你就知道</title></head> <body
_form_wrapper> <div id=lg> <img hidefocus=true src=//www.baidu.com/img/bd_logo1.png width=270 height=129> <
=1> <input type=hidden name=ie value=utf-8> <input type=hidden name=f value=8> <input type=hidden name=rsv_b
s="bg_s_ipt_wr"><input id=kw name=wd class=s_ipt value=0 maxlength=255 autocomplete=off autofocus></span><span
</div> <div id=u1> <a href=http://news.baidu.com name=tj_trnews class=mnav>新闻</a> <a href=http://www.hao12
图</a> <a href=http://v.baidu.com name=tj_trvideo class=mnav>视频</a> <a href=http://tieba.baidu.com name=tj
&u=http%3A%2F%2Fwww.baidu.com%2F%3Fbdorz_come%3D1 name=tj_login class=lb>登录</a> </noscript> <script>do
ow.location.href+ (window.location.search === "" ? "?" : "&")+ "bdorz_come=1")+ " " name="tj_login" class="lb
">更多产品</a> </div> </div> </div> <div id=ftCon> <div id=ftConw> <p id=lh> <a href=http://home.baidu.com>
ref=http://www.baidu.com/duty/>使用百度前必读</a>&nbsp;   <a href=http://jianyi.baidu.com/ class=cp-feedback>
</body> </html>
```

```
λ curl -I https://www.baidu.com
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: private, no-cache, no-store, proxy-revalidate, no-transform
Connection: keep-alive
Content-Length: 277
Content-Type: text/html
Date: Fri, 24 Jul 2020 15:34:00 GMT
Etag: "575e1f60-115"
Last-Modified: Mon, 13 Jun 2016 02:50:08 GMT
Pragma: no-cache
Server: bfe/1.0.8.18
```

curl的应用场景:

1. 浏览器是连着代理, 可以使用curl
2. 用浏览器的输入url的时候可能会携带参数, 如果用curl的话可以直接拷贝出来给朋友用的

### fetch

fetch是用来取代传统的XMLHttpRequest的。它的优点很多, 包括链式调用的语法、返回promise等。fetch api是基于promise的设计, 它是为了取代传统xhr的不合理的写法而生的。在网络上获取数据的标准接口

- 提供对请求、返回对象(标准Promise接口), fetch使用的promise对象可以使得我们使用同步的方式写异步函数。
- 提供自定义Header能力
- 提供跨域能力

```
> fetch(/xxx)
✖ Uncaught SyntaxError: Invalid regular expression: missing /
> fetch('/xxx')
< ▶ Promise {<pending>}
✖ ▶ GET https://www.baidu.com/xxx 404 (Not Found)
>
```

## postman

协作的api开发工具

## Whistle

跨平台网络调试工具(在本身的http协议上修改一些参数)

- 需要SwitchOmega插件
- node.js开发
- 支持抓包, 重放, 替换, 修改等

## 课程小结

- 简单比效率更重要(Java/HTTP等)
- 跟上时代, 掌握更多的工具