AJAX && NODE 全栈开发

什么是全栈开发?

> Full Stack developer
> 前端和后端程序都是我们一个人来开发的，并且所使用的技术语言前后端基本一致
>
> java：后台使用java来编写，前端除了使用js做各种效果，数据交互统一采用jsp技术来实现
> php：前后台都是使用php开发的，对于前端来说，数据交互依然采用的是php，只不过样式和效果还是css和js (php开源框架：think php)
> ruby、python、asp.net…
>
> javascript：目前也可以做全栈开发了，前端部分是html+css+js来完成，后端部分由node.js来完成，而node其实就是js语言，node仅仅是一个平台，能够基于v8引擎运行js代码的一个平台（和谷歌浏览器一样）

技术社区或者博客

`[stackoverflow]`

> 全世界一个非常权威专业的技术问答社区
> https://stackoverflow.com/
>
> https://insights.stackoverflow.com/survey/2016
> 2016年度全世界技术发展的总结报告

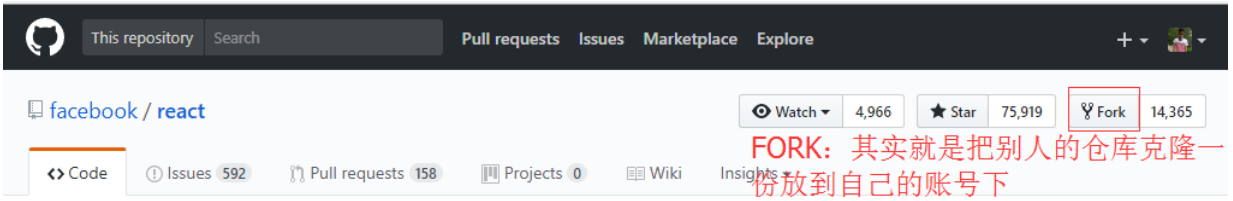
`[gitHub]`

全世界一款非常权威的开源社区(代码托管平台)，前端开发中你所知道的框架或者类库等，他们的原代码99.99999%在gitHub中都能找到

https://github.com/

以后当大家有所能力后，写一写开源的类库或者插件供别人使用，如果被人用的好，你的gitHub粉丝会越来越多，当达到一定量的时候，在国内你就寂寞了...

现在回去后，先做一件事情：把一些自己知道的框架都找到，然后把原代码fork到自己的仓库中



[codding]

中文版(国内)的gitHub，作用以及一些操作和gitHub类似

https://coding.net/

[csdn]

中国版技术问答社区，对于前端部分的资源没有那么多，偏向于后台

[其它的网站和博客]

> 掘金：https://juejin.im/
> w3cschool：http://www.w3school.com.cn/
> w3cplus：http://www.w3cplus.com/
> w3cfuns：http://www.qdfuns.com/
> web骇客：http://www.webhek.com
> css88：http://www.css88.com/
> 名人博客：廖雪峰、阮一峰、张鑫旭、大漠、崔世峰...

全栈开发的意义

前后端完全分离的项目

> 目前市场上有很多的产品都是前后端完全分离的：后台处理自己的业务逻辑(可以使用任何的技术,客户端不关心你用什么)，把客户端需要的数据内容都提供成对应的接口(接口地址)，客户端通过JS中的AJAX或者JSONP调取对应的接口，获取到数据即可，剩下的显示渲染都是前端的任务，和后台没关系

优势：

- 前后台人员是完全分开独立的，共同根据一个交互协议规范开发（API文档），而且可以同时进行开发，

提高了开发的效率

- 前端开发的内容是独立的一个站点，后台开发的内容也是一个独立的站点，项目代码是分开的，维护管理起来都非常的方便
- 可以使用AJAX等技术实现局部刷新，提高整个网站的人性化操作等

弊端：

- 使用AJAX等技术获取到数据，由JS做数据绑定(字符串拼接、ES6的模板字符串、模板引擎插件…)，不管使用哪种方式做的数据绑定，在页面的原代码中都看不到绑定的内容，不利于网站的SEO优化
- 数据渲染是由客户端渲染的，我们先请求数据，然后再渲染数据，这样无形中延长了加载的时间
- 由于AJAX属于同源请求策略，所以在开发的时候我们需要在本地配置一个和服务器相同的环境域，这样才可以调取到服务器端的数据

非完全前后端分离的项目

> 前端只需要实现样式结构以及一些JS效果，页面中的数据交互大部分都是后台语言来完成的(php、jsp、ruby…)；这样要求前端开发还需要看懂后台的部分代码；

优势：

- 数据绑定是由后台处理的，在页面的元代码中可以看到绑定的数据内容，有利于SEO优化
- 数据渲染是由服务器完成的，服务器把需要的数据以及需要展示的内容都渲染完成，然后统一返回给客户端的浏览器呈现，速度很快 `类似于京东淘宝这些网站，他们的首屏幕内容，为了能更快的加载，一般都是服务器端来渲染（后台绑定数据），其它屏幕的再交给JS请求和渲染（这样还可以减轻服务器的负担）`

网站的SEO优化

`SEO：网络推广运营`

> 假设公司有一个自己的网站或者产品，那么SEO的作用就是利用互联网的传播媒介，把公司的产品推广出去，最直接的来说：别人通过百度搜一个关键词，能不能收到你的产品，而且你产品的位置是不是第一位

为啥在百度中搜索一个关键字能查找到你的网站？

> 百度搜索引擎养了个宠物（爬虫或者叫做蜘蛛）,他会让这个宠物定期去你的网站上抓取和收录内容（此时作为网站的SEO专员，你要想尽一切办法让爬虫多收录一些关键字和内容），当用户通过百度搜索查询某个关键词的时候，百度搜索引擎内部会到它收录的词库中筛选，把所有筛选匹配到的呈现给用户，如果你的网站这个关键词收录的比别人更多，你的网站在搜索结果中就排在前面的位置

作为开发者来说，我们都可以使用什么手段，让百度多收录一些内容呢？

设置好网站的TITLE和关键词以及描述

```
<meta name="keywords" content="...">
<meta name="description" content="...">
<title>...</title>
```

如果页面中有一个LOGO，我们用H1包裹起来，把重要的关键词写在H1中，但是在样式中把文字隐藏（只给用户看图片，文字是给百度蜘蛛看的）,这样增加重点词的收录

IMG标签的alt属性中写一写关于图片的说明词，因为图片无法收录，但是alt中的文字可以收录

除此之外，我们还可以设置站点机器人、站点地图、和权重高的网站互换友情链接、软文硬文推广等技巧

重写页面的URL（伪URL重写）

SEM：百度竞价

公司在百度注册一个账号，在SEM竞价的后台，我们设置一些关键词，并且设置关键词的点击付费标准，基本上谁掏的钱多，谁排第一位

所有搜索出来的前几条中，后面带着广告两个词的，都不要随意的相信，都是花钱买的

客户端和服务器端交互的模型

> 客户端：能够向服务器端发送请求的都是客户端（一般指的都是客户的浏览器）
>
> 服务器端：能够接收到客户端的请求，并且把一些内容返回给客户端的都是服务器端（一般指的都是公司的服务器）
>
> 我们在自己电脑谷歌浏览器的地址栏中输入：https://www.baidu.com/ ，这样就能访问到百度的页面了，此时我们电脑的浏览器成为‘客户端’，百度产品所在的服务器就是‘服务器端’

职业生涯建议：开放分享

> 任何一个行业你想要有更高的建树，不仅仅是自身能力很强大，而是你的影响力很强大，通俗来讲就是有一个强大的圈子，自己在圈子中也占据这重要的位置；想要做到这一点就要‘开放分享’；
>
> - 写自己的博客：培养自己的文字功底
> - 在开源社区或者论坛分享自己的精华知识
> - 当讲师
> - 写一些小的作品（类库、组件、插件…）供别人使用
>
> …

如何把自己做好的网站(或者博客)发布出来，让别人观看？

1、先购买服务器：存储自己的项目资源文件以及一些数据内容

> 阿里云服务器 https://wanwang.aliyun.com/
> 建议购买独立主机（可以自己进入操作系统按照需要的环境以及相关配置）
> 虚拟主机一般都是管理员把东西配置好了，我们直接使用的，不是很灵活

2、把资源文件上传到服务器上

FTP上传：FileZilla
购买服务器后，会给你账号以及密码用来登录服务器，还会把服务器的外网IP地址也给你

IP地址：
当我们链接一个网络后，会自动的给我们电脑分配一个标识性的IP地址，例如：192.168.0.123，多个人链接同一个网络，大家的IP地址类似，但是也有区别，此时我们也可以说大家在同一个局域网内（这个IP地址是局域网IP或者叫做内网IP地址）
同一个局域网中：
大家的IP地址类似
网关、子网掩码、DNS都是相同的

IP地址还有一中情况：公网IP或者叫做外网IP地址，内网IP是在同一个局域网下大家可以互相的访问，但是外网IP是任何人都可以通过这个IP地址访问你的服务器

如何查看自己电脑的IP地址？
`在DOS窗口中执行：ipconfig -all`

```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . :
   描述. . . . . . . . . . . . . . . : Realtek RTL8723BE Wireless LAN 802.11n PC
I-E NIC                    链接网络后生成一个全世界唯一的标识码：MAC地址
   物理地址. . . . . . . . . . . . . : A8-A7-95-B4-54-1F
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::7147:56e4:2f83:a47e%6(首选)
   IPv4 地址 . . . . . . . . . . . . : 172.18.1.0(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.252.0
   获得租约的时间 . . . . . . . . . : 2017年9月19日 9:40:32
   租约过期的时间 . . . . . . . . . : 2017年9月19日 22:58:41
   默认网关. . . . . . . . . . . . . : 172.18.0.1
   DHCP 服务器 . . . . . . . . . . . : 172.18.0.1
   DHCPv6 IAID . . . . . . . . . . . : 145270677
   DHCPv6 客户端 DUID . . . . . . . . : 00-01-00-01-1E-33-BF-AE-50-7B-9D-A4-4D-C4

   DNS 服务器 . . . . . . . . . . . : 202.106.0.20
                                      202.106.196.115
```

3、购买域名

> 域名其实就是给不好记忆的外网IP找了一个好记的名字而已
>
> https://wanwang.aliyun.com/

4、DNS解析（域名解析）

作用：把域名和能访问服务器的外网IP关联在一起

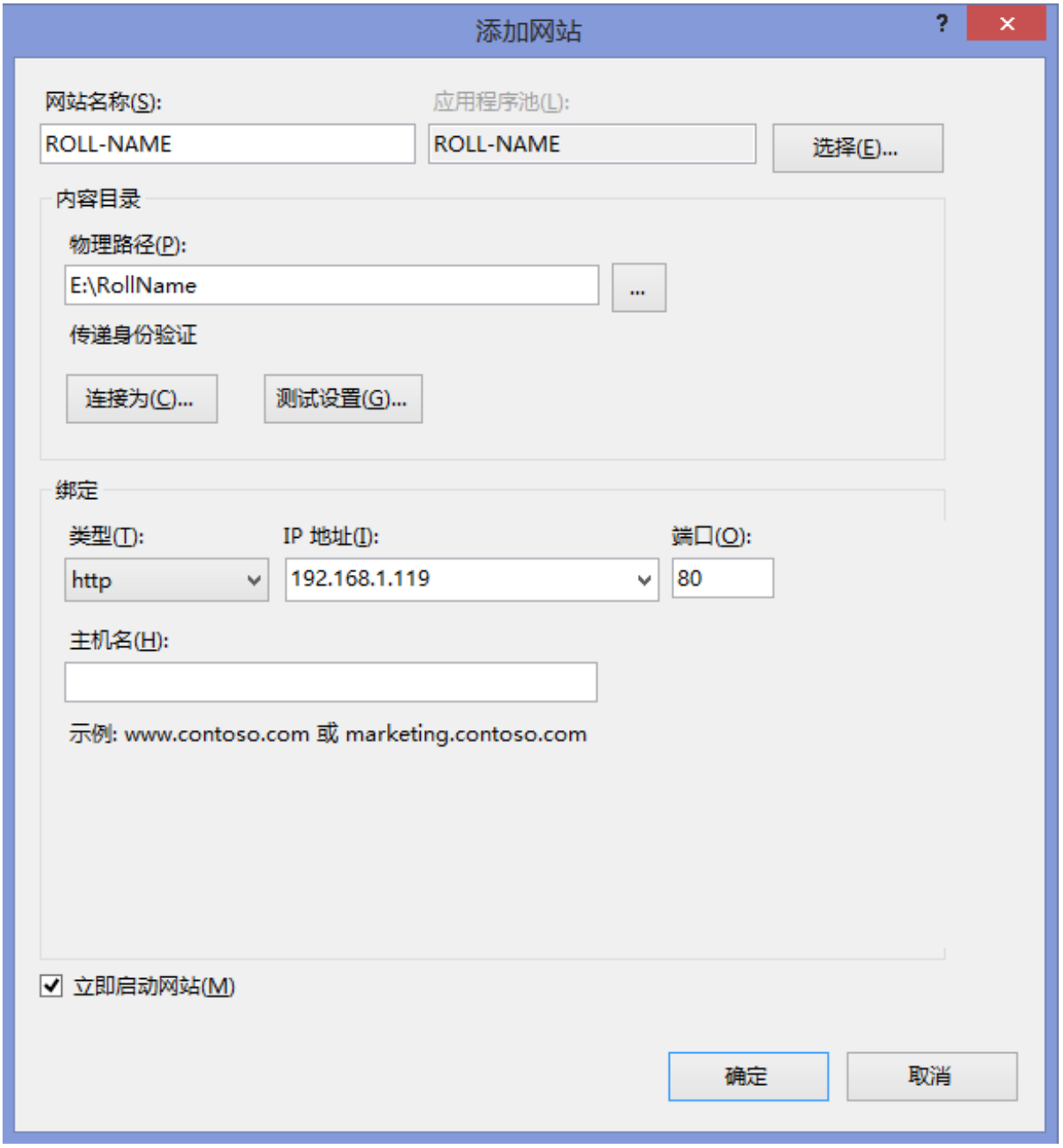
DNS服务器
全世界公用的域名解析服务器，当我们解析一个域名后，会把解析的记录存储在DNS服务器上
01 zxt.com(域名) 56.134.235.46(外网IP)

域名备案
到工信部把自己的域名和服务器IP进行登记备案，让其合法，以后自己的产品中出现乱七八糟的东西会受到法律的约束

5、以上都准备好后，我们创建服务

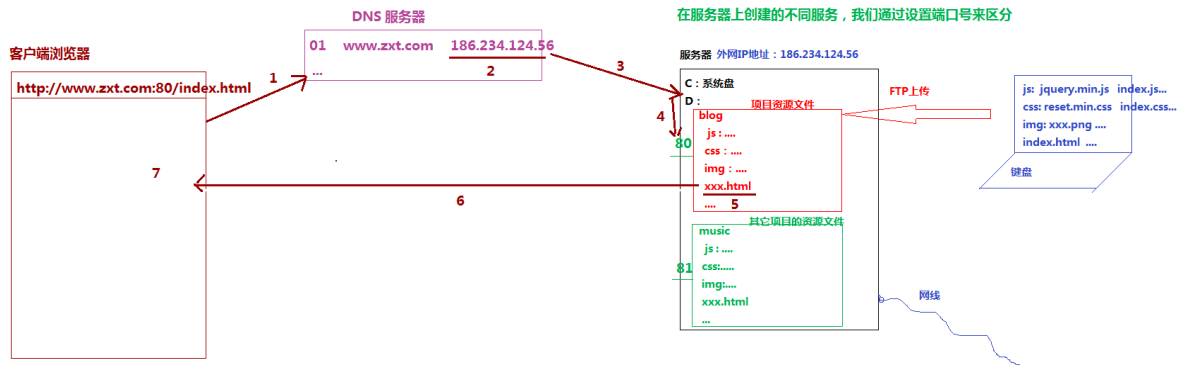
相当于招聘了几个售货员，让他们分别负责不同产品的销售，这里创建的服务会负责服务器上不同项目的管理

创建服务的工具：
iis：C#语言编写的产品一般使用IIS创建服务，服务器的操作系统一般都是windows server系统
apache：阿帕奇，PHP语言编写的产品一般都是用这个创建服务，操作系统一般是linux
nignx：这个是目前市场上最受欢迎创建服务的工具，大公司的产品一般用的都是它（可以承载更大的并发量），操作系统是linux
node：使用node.js编写的程序，我们可以基于node发布，操作系统最好也是linux
…



面试题：当我们在浏览器地址栏中输入一个网址，到最后能看到这个网址对应的页面，中间都发生了哪些事情？

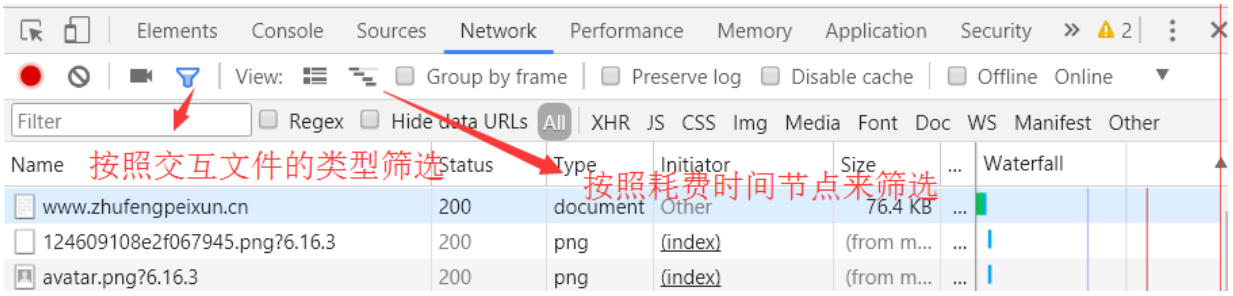
1、通过客户端输入的域名（网址），去DNS服务器上进行查找
2、在DNS服务器上找到资源服务器的外网IP地址
3、通过外网IP找到对应的服务器
4、通过客户端浏览器地址栏中输入的端口号，找到服务器上对应的服务（项目）
5、在项目中找到浏览器地址栏中请求的那个文件
6、服务器端把找到资源文件中的 原代码 返回给客户端的浏览器
7、客户端浏览器按照自己的引擎（内核）开始解析和渲染这些代码，最后呈现给用户的就是一个页面

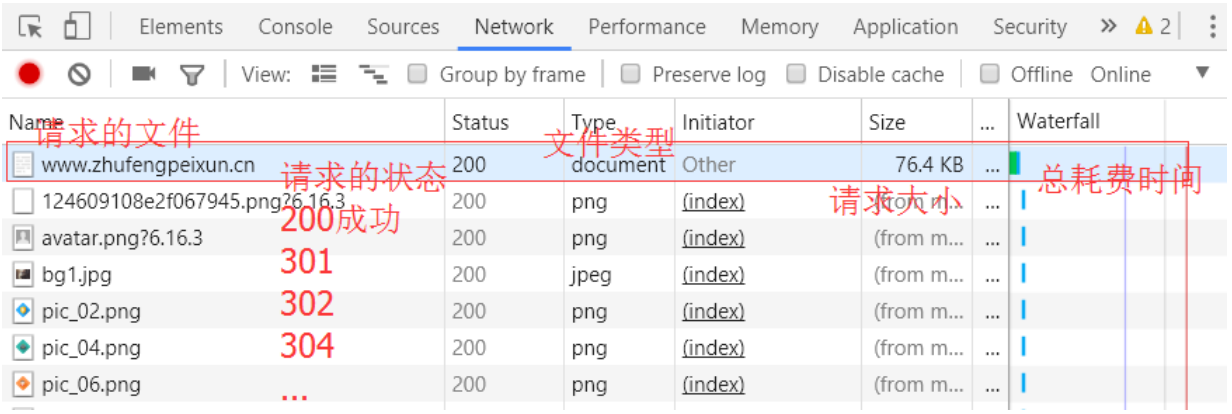


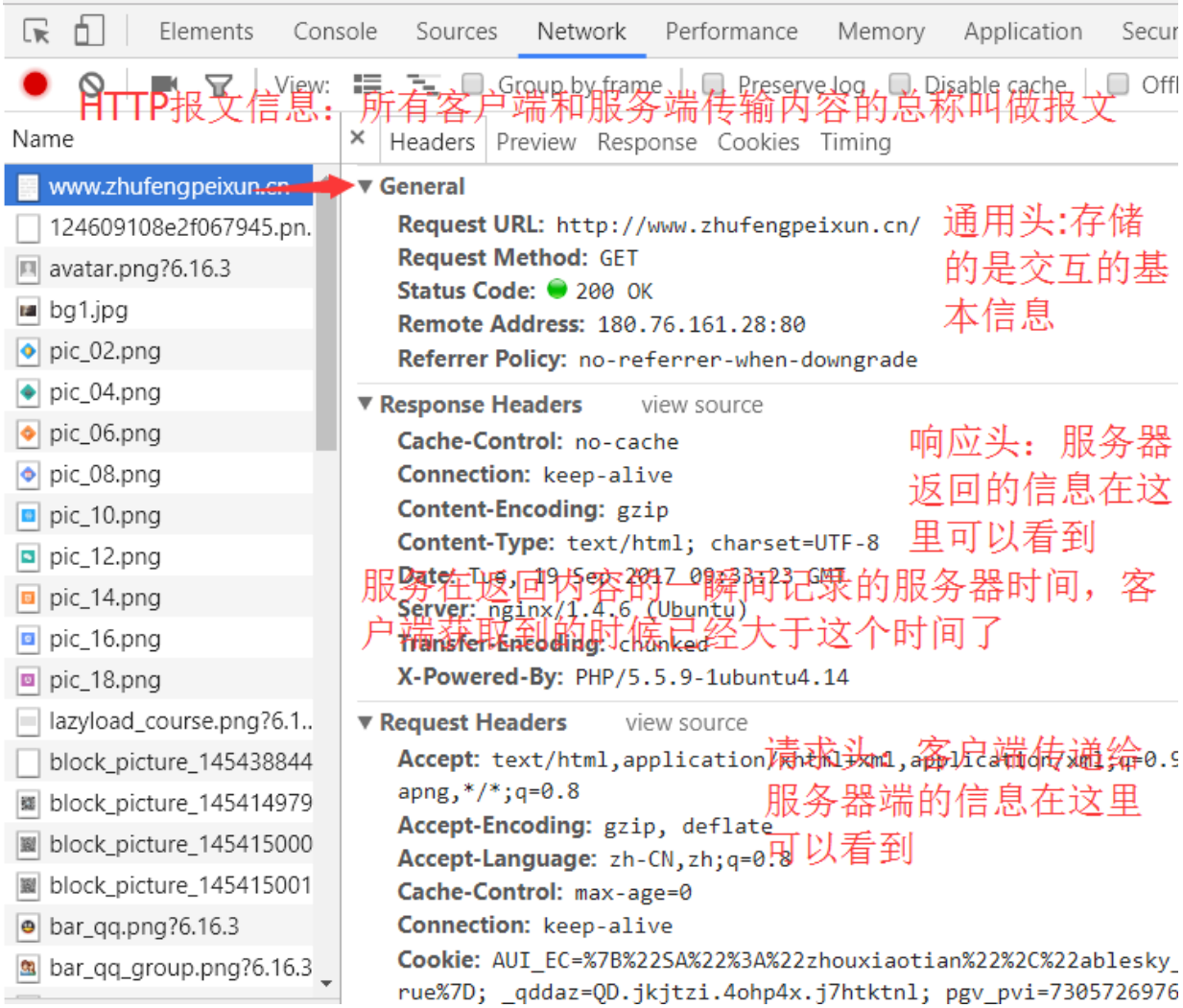
详细知识点

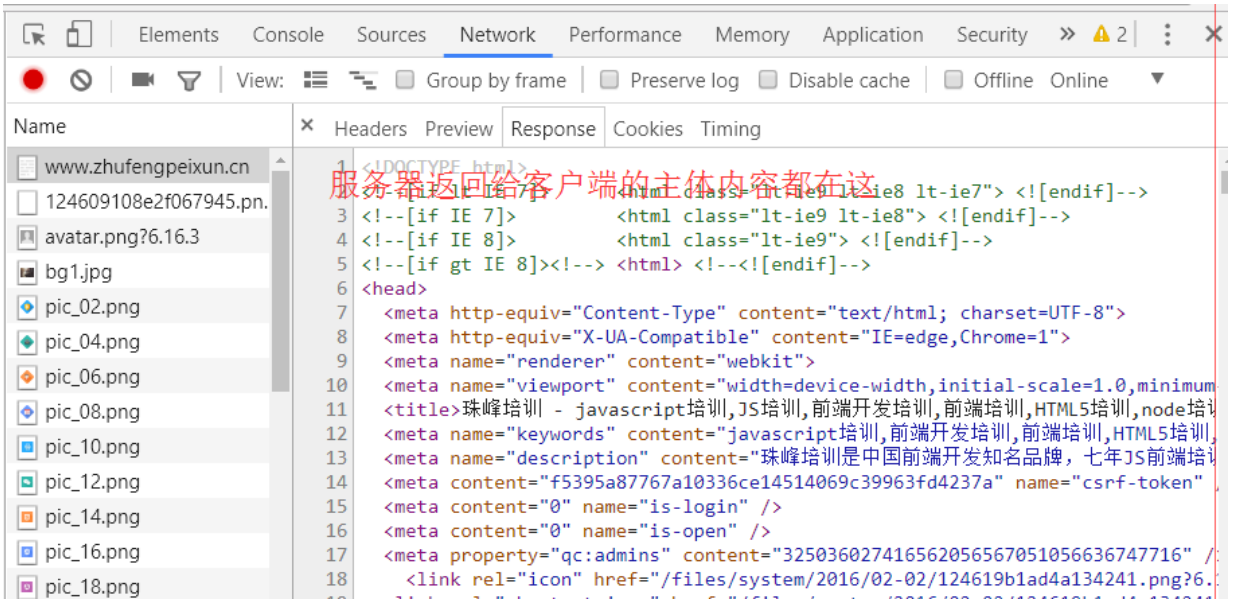
控制台：**Network**

记录了所有客户端和服务器端交互请求的信息，包括资源文件的请求以及数据的请求等等


Elements
Console
Sources
Network
Performance
Memory
Application
Security
View:
Group by frame
Preserve log
Disable cache
Offline
Online
Filter
Regex
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Name
Status
Type
Initiator
Size
Waterfall
按照交互文件的类型筛选
按照耗费时间节点来筛选
www.zhufengpeixun.cn
200
document
Other
76.4 KB
124609108e2f067945.png?6.16.3
200
png
(index)
(from m...
avatar.png?6.16.3
200
png
(index)
(from m...


Elements
Console
Sources
Network
Performance
Memory
Application
Security
View:
Group by frame
Preserve log
Disable cache
Offline
Online
Name
Status
Type
Initiator
Size
Waterfall
请求的文件
文件类型
请求的状态
200成功
301
302
304
...
请求大小
总耗费时间
www.zhufengpeixun.cn
200
document
Other
76.4 KB
124609108e2f067945.png?6.16.3
200
png
(index)
avatar.png?6.16.3
200
png
(index)
bg1.jpg
200
jpeg
(index)
pic_02.png
200
png
(index)
pic_04.png
200
png
(index)
pic_06.png
200
png
(index)


Elements
Console
Sources
Network
Performance
Memory
Application
HTTP报文信息：所有客户端和服务端传输内容的总称叫做报文
Name
Headers
Preview
Response
Cookies
Timing
www.zhufengpeixun.cn
124609108e2f067945.pn.
avatar.png?6.16.3
bg1.jpg
pic_02.png
pic_04.png
pic_06.png
pic_08.png
pic_10.png
pic_12.png
pic_14.png
pic_16.png
pic_18.png
lazyload_course.png?6.1..
block_picture_145438844
block_picture_145414979
block_picture_145415000
block_picture_145415001
bar_qq.png?6.16.3
bar_qq_group.png?6.16.3
General
Request URL: http://www.zhufengpeixun.cn/
Request Method: GET
Status Code: 200 OK
Remote Address: 180.76.161.28:80
Referrer Policy: no-referrer-when-downgrade
通用头:存储的是交互的基本信息
Response Headers
view source
Cache-Control: no-cache
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Tue, 19 Sep 2017 09:23:23 GMT
Server: nginx/1.4.6 (Ubuntu)
Transfer-Encoding: chunked
X-Powered-By: PHP/5.5.9-1ubuntu4.14
响应头：服务器返回的信息在这里可以看到
服务在返回内容的一瞬间记录的服务器时间，客户端获取到的时候已经大于这个时间了
Request Headers
view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cache-Control: max-age=0
Connection: keep-alive
Cookie: AUI_EC=%7B%22SA%22%3A%22zhouxiaotian%22%2C%22ablesky_
rue%7D; _qddaz=QD.jkjtzi.4ohp4x.j7htktnl; pgv_pvi=7305726976
请求头，客户端传递给服务器端的信息在这里可以看到



举例请求的是：http://www.zhufengpeixun.cn/

第一次客户端向服务器端发送请求，服务器端一般都会把首页面html文件中的原代码返回给客户端，客户端获取到原代码后，浏览器自上而下依次解析这些代码

解析过程中可能会遇到 `link、script、img、audio、video...` ，遇到这些标签，如果需要重新加载文件的，客户端会从新向服务器端发送请求(先到dns找外网ip，然后找到服务器，然后找到端口，在找到资源文件，最后把资源文件中的原代码返回，客户端继续渲染这些原代码)...

一个网页能正常的展示在用户面前，客户端可能需要向服务器端发送很多次的请求，请求很多文件过来才可以；如果想要提高网站的加载速度，请求的次数越少越好，所以网站性能优化最重要的一条就是：减少页面中的**HTTP**请求次数

传输协议

> REQUEST：客户端向服务器端发送请求（请求阶段）
> RESPONSE：服务器端把客户端需要的信息返回（响应阶段）
> 两个阶段都可能会有数据和内容的传输：客户端可以发送一些信息给服务器，服务器也会把一些信息返回给客户端
>
> 传输协议：用来传递客户端和服务器端交互信息的
>
> - HTTP：超文本传输协议（不仅仅能传递文本，还可以传递图片、音频、视频等资源的文件流以及xml文档流等内容）
> - HTTPS：更加安全的HTTP（传输的内容是经过SSL加密的），一般需要支付的网站都需要使用HTTPS协议，以此保证安全
> - FTP：文件上传下载传输协议

域名

域名：给服务器外网IP地址起一个好记的名字
www.qq.com：一级域名
sports.qq.com：二级域名
kbs.sports.qq.com：三级域名
…
二三级域名是不需要购买的，是自己基于当前域分出来的

.com：国际域名（商业组织或者公司）
.cn：中国顶级域名（商业组织或者公司）
.net：网络服务商（还有办公系统类的产品一般会用 这种域名）
.org：机构组织域名（非盈利的）
.gov：政府官方域名
.edu：教育院校域名
…

端口号

端口号：在同一台服务器上区分不同的服务以及项目，我们使用不同的端口号即可

同一台服务器端口号的取值范围：0~65535

我们在电脑上安装的一些软件，大部分都会占用一下当前电脑的某个端口号，例如：迅雷...

平时上网我们不需要手动输入端口号，不同的传输协议都有自己默认的端口号（我们自己不写，浏览器会主动为我们增加上对应的端口号）
HTTP -> 80
HTTPS -> 443
FTP -> 21

请求资源文件的路径名称

/student/index.html
请求当前项目中 student 文件夹中的 index.html 页面，但是需要注意的是：当前URL可能是伪造的，仅是为了增加SEO的收录和抓取

问号传参

?name=sh&age=18
URL问号后面传递的值统称为‘问号传参’：
xxx=xxx&xxx=xxx...

作用：
1、客户端可以通过问号传递参数的方式，把一些数据和内容传递给服务器，服务器接收到内容后可以做相关的处理
2、用于两个页面之间的数据传输，例如：列表页中有很多数据，点击每一条数据都可以跳转到详情页面（详情页面是一个），但是需要在同一个详情页面展示不同的信息（列表页点击的是哪一个就展示谁的信息）；我们在点击跳转的时候，把点击的信息通过问号传递参数的方式传递给详情页面，进入到详情页面后，首先获取地址栏中问号传递的参数值，通过不同的值展示不同的信息；

HASH值

> #pay
> HASH(哈希)值：在一个URL中#后面的值被称为哈希值
>
> 作用：
> 1、也可以通过这种方式把数据传递给服务器端或者传递给另外一个页面（类似于问号传参：这种模式不常用）
> 2、锚点定位
> 3、前端路由

`URI/URL/URN`

> http://www.zhufengpeixun.cn:80/student/index.html?name=sh&age=18#pay
> URI：统一资源标识符（URL+URN）
> URL：统一资源定位符
> URN：统一资源名称

前端优化的一些方法和建议

[减少HTTP的请求次数]
1、CSS SPRITES：雪碧图，把图片压缩合并在一张大图上，背景图片统一导入大图，使用background-position定位到具体的小图

```
/*把导入大图提取为公共的样式，只写一次导入，结构中需要的话，增加这个样式类即可*/
.bg-img{
    background:url('大图地址') no-repeat;
}
.box1{
    background-position:xxx xxx
}
.box2{
    background-position:xxx xxx
}

<div class='bg-img box1'></div>
<div class='bg-img box2'></div>
```

2、把CSS合并成为一个，把JS也合并成为一个，减少CSS以及JS等文件的请求次数（真实项目中，我们使用gulp、webpack这些自动化工具，可以完成合并压缩）

3、在移动端开发中，如果页面中的CSS和JS不是很多的话，尽量使用内嵌式，以此减少HTTP请求次数

4、图片的BASE64（慎用，我们在webpack中可以统一设置一下，把多大范围内的图片都BASE64了）

5、把不需要经常更新的数据或者资源，进行304缓存或者进行本地存储(localStorage)，减少在一定时间段内重复向服务器发送HTTP请求（在此阶段内我们只需要读取缓存或者本地的数据即可）

6、为了加快页面首次打开速度，我们把图片和数据都做一下延迟加载；数据的延迟加载(异步加载)：滚动到底部加载更多数据、分页效果…

7、音视频等资源文件，开始的时候设置preload='none'，也就是开始加载页面的时候不加载资源文件，只有当播放的时候再加载

[减少请求资源文件的大小]

1、图片资源、JS/CSS文件资源等都要进行压缩：图片压缩的前提是保证图片观看的质量

2、减少每一次AJAX请求数据量的大小，对于大数据处理，我们尽量分多次请求（AJAX异步加载或者分页等操作就是这个原理）

3、开启服务器的GZIP压缩，不仅把文件都压缩了，而且传输的数据和内容也是经过压缩的（这个操作由服务器人员来完成）

4、减少页面中的冗余代码，提高代码的重复利用率，例如：JS中的函数封装、CSS中公共样式提取等都是在做这个事情

5、在项目中如果使用框架或者类库，需求允许的情况下，尽量使用轻量级的（尤其是移动端开发）

6、对于图片等资源的交互传输，我们尽量把图片等富媒体资源转换为二进制编码，使用二进制文件流的形式传输，不仅加快了文件传输的速度，也节省了传输的大小

[CSS代码优化]

1、外链式导入CSS样式表，我们使用link代替@import，@import是同步导入，当前资源文件没有加载完成，下面的CSS也不会执行

2、把CSS放在页面的头部，优先加载和渲染；JS放在底部，等待最后在加载渲染；

3、能够使用CSS解决的，绝对不使用JS或者FLASH

4、避免使用CSS表达式

5、少使用filter滤镜：IE低版本浏览器中，filter会阻止图片的渲染，使用滤镜消耗的性能较大

6、尽量使用transform变形来代替传统css样式的改变，因为transform开启了硬件加速，性能流畅一些

7、修改元素的样式尽量是修改样式类名，少使用style修改

8、CSS中尽量多使用样式类选择器，少使用通配符或者标签选择器；少使用单规则属性选择器等；（因为CSS选择器是从右向左进行匹配的）

```
1. .box a{}
2. 先找到所有的A标签,然后再匹配筛选在.box下的a
3.
4. .box .link{}
5. 先找到所有样式类为link的(这样肯定比使用标签名找到的少很多)
6.
7. a[href='xxx'] 先找到所有的href，然后属性值为xxx，再向上匹配它的元素
```

9、避免深层次节点嵌套

雅虎优化的35条建议：总结更多的CSS代码优化技巧

[JS代码优化]

1、减少闭包的使用（闭包会产生不销毁的作用域，占用了浏览器很多的内存）

2、一定要养成手动释放堆栈内存的习惯，经常去清空一些没有用的内存（例如：手动清除没用的定时器、把一些无用的对象赋值为null…）

3、避免内存互相引用嵌套导致的内存泄漏问题（高程三后面有关于内存泄漏的讲解）

4、不要使用with，少使用eval，少使用document.write以及alert等（页面中的提示框，我们可以自己编写一个组件来完成）

5、注意DOM的回流重载问题

6、少使用同步编程，多用异步编程，注意页面中不要出现死循环

7、减少浏览器的异常信息抛出，最好使用TRY CATCH把它捕获了

8、少使用全局变量，依托单例、发布订阅、promise等设计模式，来规划我们的代码结构，保证代码的严谨和有利于维护

9、事件绑定尽量使用事件委托来处理

10、减少作用域链的查找（作用域之间不要嵌套太深）

11、优化代码逻辑，用最简单的方式来实现一个功能需求，一些没必要的判断或者没必要执行的代码都让其不执行即可（一些编程的好习惯）

12、为了保证安全，一些重要的数据信息在进行交互的时候要进行加密，而且传输方式尽量使用POST方式，GET请求不安全

13、前后端的数据交互，数据格式最好是JSON格式的：结构清晰明了，处理起来方便，传输性能也比较的不错…

…

[有助于SEO优化推广的]

1、设置合理的 title、keywords、description等信息

2、标签语义化

3、增加页面中关键词的曝光率，把一些重要的关键词多放在权重较高的标签中（例如：H1中有图片和文字，文字是给爬虫看的，图片是给用户看的…）

4、img设置alt属性，属性值是对图片描述的关键词，让爬虫收录图片信息

5、页面中千万不要出现死链接和空链接

6、避免页面301和302重定向，重定向不太利于网站权重和排名的上升

7、动态页面静态化（伪URL重写）

8、页面中尽量不要使用iframe，不仅不利于SEO优化，而且性能也会受到影响

…

[其它优化]

1、后台优化：一个网站性能优化前端是次要的，后台优化（尤其是数据分析挖掘的优化）比前端重要很多，后台优化可以把页面加载速度提高很多

2、服务器优化：不管前后端怎么优化，页面整体加载速度只会提升，但是不会特别的明显，尤其是访问人数过多的时候，此时服务器优化很重要：

- 加服务器
- CDN地域分布式