### 客户端与服务器

DNS：域名解析器Domain Name Server

端口：如果把IP地址比作一间房子 ，端口就是出入这间房子的门。真正的房子只有几个门，但是一个IP地址的端口可以有65536（即：2^16）个之多！

80端口是为[HTTP](http://baike.baidu.com/view/9472.htm" \t "http://baike.baidu.com/item/_blank) HyperText Transport Protocol 即[超文本传输协议](http://baike.baidu.com/view/468465.htm" \t "http://baike.baidu.com/item/_blank)开放的，此为上网冲浪使用次数最多的协议，主要用于WWW World Wide Web即万维网传输信息的协议。可以通过HTTP地址加“:80”来访问网站，因为浏览网页服务默认的[端口号](http://baike.baidu.com/view/642103.htm" \t "http://baike.baidu.com/item/_blank)都是80，因此只需输入网址即可，不用输入“:80”了。

通讯协议：http / https / SMTP / IMAP / POP3，http / https是网页协议。IMAP和POP3协议是最广泛用于在电子邮件客户端应用程序中接收邮件通信的协议。

静态页面：页面就是一个完整的html；

动态页面是一个html，页面的内容由js生成。

静态网站和动态网站：服务器的能力。静态网站服务器不做任何事情，只是简单地将请求页面发送给浏览器。一般来说，网址上以.html结尾的都是静态网站，这个html是提前写好的（包括图片、媒体文件等静态资源）。

缺点：随着网站规模的增大，可维护性逐渐降低，且没有交互性。

动态网站是指服务器接收用户的请求和参数，根据参数在服务器端执行相应的代码（php, c#, java...），拼接字符串动态合成html文本，并将其返回给浏览器，这里动态生成的不一定是完整的页面，可能仅仅是页面的一部分，或者仅仅是数据。

### HTTP协议

1. 概述：

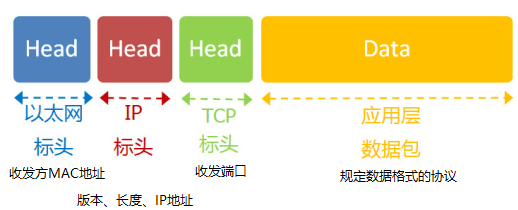
http（超文本传输协议）是一个基于请求与响应模式的、无状态的、应用层的协议，常基于TCP的连接方式。

http请求由三部分组成：请求行、消息报头、请求正文。

HTTP响应也是由三个部分组成，分别是：状态行、消息报头、响应正文。

HTTP长连接，与一般每次发起http请求或响应都要建立一个tcp连接不同，http长连接利用同一个tcp连接处理多个http请求和响应，也叫HTTP keep-alive，或者http连接重用。使用http长连接可以提高http请求/响应的性能。

1. HTTP协议的主要特点可概括如下：
2. 支持客户/服务器模式。
3. 简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。
4. 灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。
5. 无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。
6. 无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。



|  |  |
| --- | --- |
| 应用层  Application layer | 作用：规定应用程序的数据格式。  动态主机配置协议DHCP Dynamic host configuration protocol |
| 传输层  Transport layer | 作用：建立“端口到端口”的通信。  UDP User Datagram Protocol用户数据报协议在数据前加上端口。  TCP Transmission Control Protocol 传输控制协议有确认机制。 |
| 网络层  Network layer | 作用：建立“主机到主机”的通信  IP Internet Protocol地址帮助我们确定计算机所在的子网络，MAC地址则将数据包送到该子网络中的目标网卡。  子网掩码subnet mask能判断任意两个IP是否处在同一子网络。同一个子网络传输用广播，否则用route. |
| 链接层  Link layer | 以太网Ethernet规定，一组电信号构成一个数据包，叫做"帧Frame"。帧分成两个部分：标头Head（18字节）和数据Data（1500字节）。标头包括收发双方的MAC地址。  通过ARP Address Resolution Protocol 地址解析协议可知同一子网络内的主机MAC地址。 |
| 实体层  Physical layer | 把电脑连接起来，负责传送0/1电信号 |

### wamp集成环境

windows操作系统、Apache提供静态资源服务（html, js, css, img），MySQL数据库、php编程语言，可以用来开发网站。（.php文件也放在服务器上）。

服务器的含义有两层：一是后台的大电脑；二是Apache这个软件。

1. Apache

Apache取自“a patchy server”的读音，意思是充满补丁的服务器，因为它是自由软件，不断有人来为它开发新的功能、新的特性来修改原来的缺陷。

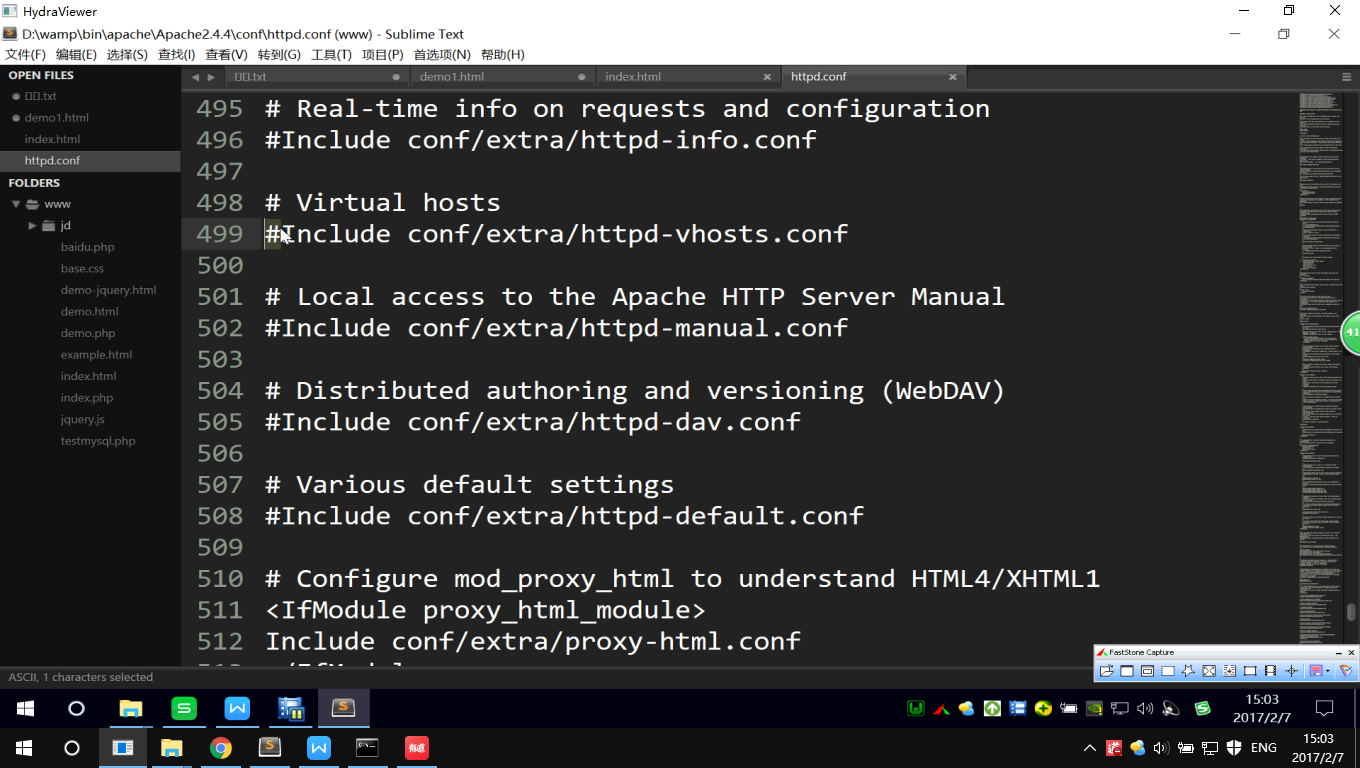
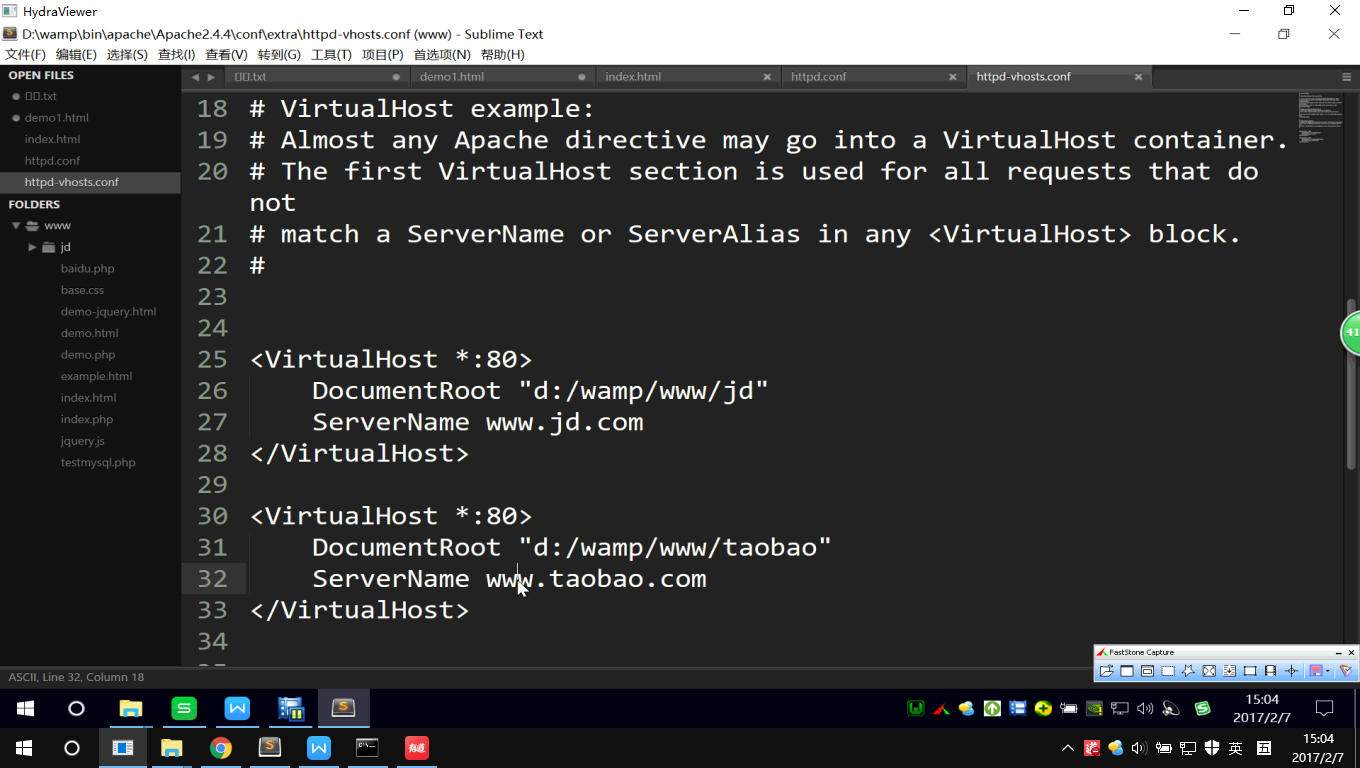
1. MySQL

MySQL是一个关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所用的SQL语言是用于访问数据库的最常用标准化语言。Linux作为操作系统，Apache或Nginx作为Web服务器，MySQL作为数据库，PHP/Perl/Python作为服务器端脚本解释器就可以可以建立起一个稳定、免费的网站系统，被业界称为LAMP/LNMP组合。

### 怎样让别人访问到自己的页面？

1. 找到Apache -- httpd.conf文件，打开，或者找到D:\wamp\bin\apache\Apache2.4.4\conf\httpd.conf拖进编辑器打开
2. 把268行的Deny改成Allow from all
3. Restart All Services
4. Window + R运行，输入cmd，然后输入ipconfig /all，在以太网里面找IPv4地址即可找到自己的IP地址
5. 在浏览器的地址栏中输入IP/index.html
6. 如果www文件夹不在默认的wamp下，那么在httpd.conf中找到DocumentRoot，把它后面的地址”d:/wamp/www”改一下，然后重启。

### 怎样让Apache支持多个站点？

1. 在httpd.conf中找到Virtual hosts，把其下面的499行的#去掉；
2. 找到499行的文件：wamp/bin/apache/Apache2.4.4/conf/extra/httpd-vhosts.conf，打开；
3. 找<VirtualHost \*:80>，只留下
4. 找项目的根目录，修改DocumentRoot 和ServerName

<VirtualHost \*:80>

DocumentRoot "D:\wamp\www\tom"

ServerName www.tom.com

</VirtualHost>

1. 打开C:\Windows\System32\drivers\etc\hosts
2. 在最后一行修改域名和IP地址的映射关系

127.0.0.1 www.wjs.com

127.0.0.1 www.tom.com

1. Restart All Services

请求有很多种，在浏览器地址栏中输入域名，请求的是html页面；

输入.json也照样能输出json数据。

Ajax是诸多请求中的一种，它请求的是.json/.xml/文件，当然也可请求html(例iframe)。

2005年，Adaptive Path公司的Jesse James Garrent发明了Ajax这个词，用于概括异步加载页面内容的技术。传统的网页（不使用 Ajax）如果需要更新内容，必须重载整个网页页面。使用Ajax只更新页面的一小部分。

Ajax技术的核心是XMLHttpRequest对象，它充当着浏览器的脚本（客户端）与服务器之间的中间人的角色。以往的请求都由浏览器发出，而JavaScript通过这个对象可以自己发送请求，同时也自己处理响应。

微软的Outlook Web Access 应用了 Ajax 技术，并取得了成功。2005 年初，许多事件使得 Ajax 被大众所接受。Google 在它著名的交互应用程序中使用了异步通讯，如[Google](https://baike.baidu.com/item/Google" \t "https://baike.baidu.com/item/ajax/_blank)、Google 地图、Google 搜索。

**Ajax是什么？无页面刷新。可不可以跨域？不可以，但是可以用jsonp (json + padding) 解决。**或者用document.domain+iframe / location.hash+iframe / window.name+iframe / window.postMessage / flash插件

$\_GET侧重于向服务器索要数据，比如发送“学号=2006205”，服务器会返回各科成绩；

$\_POST侧重于向服务器给予数据，比如把注册信息（姓名、性别、年龄、身高、体重）全部发送给服务器，服务器返回“保存成功”。

|  |  |
| --- | --- |
| $code = $\_GET["code"];  http://www.test.com/submit.php?code=125 | get只能从php的url获取数据，在表单中输入的无效 |
| <form action="practise.php" method="post">  $code = $\_POST["username"]; | post可以获取form里输入的内容  [http://www.test.com/submit.html](http://www.test.com/test.html)  所以表单提交都用post |

MIME(Multipurpose Internet Mail Extensions)多用途互联网邮件扩展类型。是设定某种[扩展名](http://baike.baidu.com/view/1080.htm" \t "http://baike.baidu.com/view/_blank)的[文件](http://baike.baidu.com/view/345685.htm" \t "http://baike.baidu.com/view/_blank)用一种[应用程序](http://baike.baidu.com/view/330120.htm" \t "http://baike.baidu.com/view/_blank)来打开的方式类型，当该扩展名文件被访问的时候，[浏览器](http://baike.baidu.com/view/7718.htm" \t "http://baike.baidu.com/view/_blank)会自动使用指定应用程序来打开。

设置服务器响应的文件类型：

header(“Content-Type:text/plain;charset=utf-8”); //把返回的数据解释成纯文本

header(“Content-Type:text/html;charset=utf-8”); //把返回的数据解释成页面

{ ".txt", "text/plain" }

两种不同的业务逻辑：

1. 数据验证：

输入用户名时要时时验证这个用户名是否已存在，此时是异步请求，需要用Ajax。

<form>

<input type="text">

<input type="password"/>

</form>

此时<form>不需要有action和method属性，<input>也不需要有name属性，因为Ajax中会传递相关参数。真实的业务逻辑是：<input type="text">失焦时，发送Ajax请求。

1. 数据提交

<form action="1.php" method="post">

<input type="text" name="username"/>

<input type="password" name="password"/>

<input type="submit" value="submit"/>

</form>

"submit"是同步请求，action后面是提交的地址，method后面是传递数据的方式。"1.php"会获得表单中输入的数据。按"submit"按钮页面会跳转到"1.php"。

客户端发送Ajax请求时必须包括三部分、地址(url)、类型(get/post)、参数

地址是：客户端想要请求什么文件，如js / php / json / xml

如果请求的是js / css / json文件，直接返回这些静态文件；

如果请求的是php，那么Apache就用它里面的php解析器来解析相应的php文件，然后把浏览器可以识别的代码返回，再经渲染呈现出来。

发送Ajax的步骤：

1. 创建XMLHttpRequest对象

var xhr = new XMLHttpRequest;

1. 准备发送

xhr.open(“get”,”url地址”,true); //第一个参数可以是get/post

如果是get请求，那么请求参数必须在url中传递，encodeURI()用来对中文参数进行编码，防止乱码。

1. 执行发送动作

|  |  |  |
| --- | --- | --- |
| **get** | xhr.send(null); | 参数要写null |
| **post** | xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded");  xhr.send(parameter); | 参数通过send()传递，因为设置了请求头，所以参数不需要encodeURI()编码 |

get从指定的资源请求数据；post向指定的资源提交要被处理的数据。

1. 指定回调函数

xhr.onreadystatechange = function(){

//该函数调用的条件是readyState状态发生变化（不包括从0变为1）

if(xhr.readyState == 4 && xhr.status == 200){

alert(xhr.responseText);

}

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| readyState | 0 | 1 | 2 | 3 | 4 |
| 表示 | xhr对象创建完成 | 已发送了请求 | 浏览器已经收到了服务器响应的数据 | 正在解析数据 | 数据已经解析完成，可以使用，但不一定是正常数据 |

|  |  |  |  |
| --- | --- | --- | --- |
| http的常见状态码 | 200 | 404 | 500 |
| 表示 | 响应成功 | 没有找到请求的资源 | 服务器端错误 |

数组也属于json的范畴，是json形式的一个特例。

### 跨域

1. 静态：通过<script src = “ ”></script>的src属性；默认是同步的。但是动态创建的<script src = “ ”></script>标签发出的请求是异步请求。
2. 动态：jsonp的本质动态创建<script></script>标签，然后通过src属性发送跨域请求，服务器端响应回来的是【函数调用】（其实就是一段js代码）。在发送请求之前必须先声明一个函数，函数的名字和该函数的参数都要用？拼接到url地址后面。
3. jQ中Ajax的作用：告诉服务器一个门牌号（data: {param：””}）和一项重要任务（即success: function(){}），服务器根据门牌号（data里传递的参数）先找到这个人（即在后台取到的data），并给任务取个名字（比如cb），最后让人去执行任务（即调用success后的回调函数cb (data)）。

### 模板引擎 artTemplate

1. 嵌入子模板

{{include 'template\_name'}}

{{include 'template\_name' news\_list}}

这两种方式母、子模板所用的数据都**在一个data对象里**面，不同之处在于它们所用的数据的层级。如果是同一个层级就只写'template\_name'即可，如果层级不同就要用news\_list指明。

<script type="text/html" id="temp1"> //这是母模板

{{each list}}

<div>

<h2>{{$value.title}}</h2>

{{include 'temp2' $value.images}}

</div>

{{/each}}

</script>

<script type="text/html" id="temp2"> //这是子模板

<div>

<ul>

{{each list}}

<li>{{$value.image}}</li>

//如果这里遍历的是$value.id就只写{{include 'temp2'}}即可。

{{/each}}

</ul>

</div>

</script>

<script type="text/javascript">

$(function(){

var data = {

list:[

{

title:"这是一个测试标题1",

id: "1",

contents:[

{"content":"这是一段内容11"},

{"content":"这是一段内容12"},

{"content":"这是一段内容13"}

],

images:[

{"image":"这是一张图片11"},

{"image":"这是一张图片12"},

{"image":"这是一张图片13"}

]

},

{

title:"这是一个测试标题2",

id: "2",

contents:[

{"content":"这是一段内容21"},

{"content":"这是一段内容22"},

{"content":"这是一段内容23"}

],

images:[

{"image":"这是一张图片21"},

{"image":"这是一张图片22"},

{"image":"这是一张图片23"}

]

}

]

};

var html = template('temp1', data);

$("#main").html(html);

});

</script>

第一次Ajax请求回来主列表。

点击箭头，发送第二次Ajax，请求回来副列表。

这时就不能用嵌入子模板的方式，因为母、子模板只能循环遍历一个data对象，即第一次Ajax数据。

要想让第二次Ajax数据渲染到页面，得使用template.compile( ).

1. [在javascript中存放模板](http://aui.github.io/artTemplate/demo/compile.html)

compile的作用是根据source这个字符串模板生成一个render**函数**，也就是说compile方法的返回值是一个函数。render可以接收一个参数（即传入的数据），将数据放进模板中，再生成静态的html片断。

其实跟template方法是类似的，都是把数据放入模板进行循环。只是这个模板存放于<script></script>中，而不是单独的模板文件<script id = “tem” type = “text/template”></script>

<script>

var source = '<ul>'

+ '{{each list as value i}}'

+ '<li>索引 {{i + 1}} ：{{value}}</li>'

+ '{{/each}}'

+ '</ul>';

var render = template.compile(source);

var html = render({

list: ['摄影', '电影', '民谣', '旅行', '吉他']

});

document.getElementById('content').innerHTML = html;

</script>

location是BOM对象的一个属性，它也是一个对象，它有若干属性

|  |  |
| --- | --- |
| 属性名 | 属性值 |
| location.hash | "#/active" |
| $location.url() | /active 也是字符串，但是没有# |
| location.href | "file:///D:/wamp/www/test/todos/index.html#/" |
| location.search | "?para=5806" 用于获取参数 |

**[JS获取url参数及url编码、解码](http://www.cnblogs.com/yanhaidong/archive/2010/07/15/2339058.html)**

在js中可以使用escape(), encodeURL(), encodeURIComponent()，三种方法都有一些不会被编码的符号：

escape()：@ \* / +

encodeURL()：! @ # $& \* ( ) = : / ; ? + '

encodeURIComponent()：! \* ( ) '

在java端可以使用URLDecoder.decode(“中文”, "UTF-8");来进行解码

但是由于使用request.getParameter（）来获取参数时已经对编码进行了一次解码，所以一般情况下只要在js中使用

encodeURIComponent（"中文"）;

在java端直接使用request.getParameter（）来获取即可返回中文。

如果你想在java端使用URLDecoder.decode(“中文”, "UTF-8");来解码也可以在js中进行二次编码，即：

encodeURIComponent（encodeURIComponent（"中文"））;

如果不进行二次编码的话，在java端通过decode方法取的会是乱码。