第一部分：了解web，网络基础及HTTP协议

第二部分：http协议详解

第三部分：HTTP协议的相关扩展

biny1106@163.com

Web基础15%，网络基础30%，HTTP协议55%

1. Web基础
2. 在网页地址栏输入url时，web页面是如何显现的，其中的过程(较重要)

以后再说，这是一个大问题，最后就是要弄懂这道题

1. http为了啥诞生

1990年出的HTML0.9版本，1996年出的1.0版本，1997年出的1.1版本一直用到现在，现在新出2.0版本，但没被推广开来，

为了知识共享性而诞生，借助多文档之间相互关联形成的超文本，连接可相互阅读的www(万维网，World wide web)

·HTML作为页面的文本标记语言

·HTTP作为文档间的传递协议

·URL指定文档所在地址

·93.NCSA浏览器鼻祖

1. HTTP协议是TCP/IP协议族(几十种以上的协议)的子集

HTTP tcp UDP ICMP等等，分层管理(便于维护)：

结构：应用层，传输层，网络层，数据链路层

应用层：向用户提供应用服务时通信的活动(HTTP属于应用层的面对对象的协议)

传输层：对于上层应用层，提供出于网络连接中的两台计算机之间的数据传输(TCP协议)

网络层：用来处理网络上流动的数据包(数据包是最小的网络传输单位)

链路层：用来处理连接网络的硬件部分

客户端 服务器端

1、应用层

HTTP

HTTP

TCP

2、传输层

TCP

3、网络层

IP协议

IP协议

4、链路层

网络

网络

通信传输流

客户端和服务器端通过链路层的网络相连，客户端发送的一个请求传到服务器端，再返回来

1. IP，TCP和DNS

·IP协议(负责数据传输)：位于网络层，IP和IP地址完全不同，IP是个协议

·IP地址：指明节点被分配的地址

·TCP协议：确保可靠性(用三次握手策略)

三次握手：1.先发送一个带有SYN标志的数据包给对方，2.接受端收到后会回传一个带有SYN/ACK标志的数据包来表示传达确认信息，3.最后发送端再回传一个带有ACK标志的数据包，代表握手结束

示意图：

知道，已经收到。并发标有ACK的数包

收到你给我发送的数据包！发送标有SYN/ACK的数据包

发送标有SYN的数据包

接受端

发送端

·DNS：负责域名解析，和HTTP一样，是位于应用层的协议，提供域名到IP地址之间的解析服务。IP地址是给电脑看的，客户端发送一个域名请求，DNS解析返回一个IP地址，通过这个IP地址访问这个域名的web服务器通

[www.baidu.com](http://www.baidu.com) --> 123.234.345

1. URI和URL

·URI:统一资源标示符

作用：用于表示某一互联网资源名称的字符串

格式：协议的方案名/登录信息(不必要)/服务器地址/服务器端口号(不写会自动接到一个端口)/带层次的文件路径/查询字符串(不必要)/片段表示符(不必要)

例：[http://()user:pass()@www.58().jp:](http://()user:pass()@www.58().ip:)80/()dir/index.html

()?Uid=1()#ch1()

·URL:统一资源定位符

作用：表示资源的地点(表示互联网上所处的位置)

格式：协议的方案名/服务器地址/带层次的文件路径/各类参数

例：http://()www.58.jp/()dir/index.html()?Uid=1()

URL是URI的子集

1. 各个协议间的关系

如图

简单的http协议(HTTP使用很简单)解析

1. HTTP用于客户端和服务器端的通信

客户端：指与服务器端相对应，为客户端提供本地服务的程序，另一种说法也叫，请求访问文本或图像等资源的一端成为客户端

服务器端：提供资源响应的一端为服务器端

1. 通过请求和响应的交换达成通信

客户端的请求：GET/ index.html HTTP/1.1 Host:hackr.jp

请求报文构成(能看懂就行)：方法/ 资源对象(URL) 协议/版本 域名或IP 请求首部字段 内容实体(由报文主体编码后形成)

意思：请求访问某台HTTP服务器上的index.html的页面资源

响应报文：HTTP/1.1 200 OK ..... <html>

响应报文构成：协议版本，状态码，用以解释状态码的原因短语，可选的响应首部字段，实体主体构成(由报文主体编码后形成)

1. HTTP是不保存状态的协议，为了更快的处理大量数据，用Cookies储存一些东西包括一些状态，Cookies会从服务器端发来的响应报文内的一个Set-Cookie的首部字段信息，通知客户端保存Cookie，下次发送请求时会加上Cookie值，服务器端会通过这个Cookie检查哪一个客户端发来的请求，对比服务器上的记录，最后得到之前的状态信息
2. 请求URI定位资源

HTTP方法

1. GET:获取资源

2.POST:传输实体主体(隐式的GET)

3.PUT:传输文件 任何人都可以上传，存在安全隐患，如果一定要使用需配合验证程序

4.HEAD:获得报文首部 5.DELETE:删除文件 6.OPTIONS:询问支持方法 7.TRACE:追踪路径8.CONNECT:要求用隧道协议连接代理

1,2很重要，其他的很少用，这些方法都是HTTP1.1支持的

1. 持久连接节省通信量：建立一次TCP连接后进行多次请求和响应的交互，只要任意一端没有明确提出断开连接，则保持TCP连接状态，减少了TCP重复连接断开所造成的额外开销，减轻服务器负载
2. 持久连接的出现使得管线化的方式成为了可能，管线化：将HTTP请求整批提交的技术，而在传送过程中不需先等待服务端的回应，因为管线化需要持久连接的支持，所以只有HTTP1.1支持，并且只有GET和HEAD可以用，POST有限制

如果有问HTTP1.0和1.1版本有什么区别，可以说1.0没有持久连接也就没有管线化

HTTP详解

HTTP状态码 HTTP报文信息 HTTP首部字段

1. HTTP状态码：用来描述返回的请求结果，记住状态码，我们可以知道是正确的处理了请求还是出现了错误，状态码响应类别一共有五种，比如200；数字中的第一位指响应类别，后两位无分类。

五种类别分别为： 原因短语：

1XX(信息性状态码) 接受的请求正在处理 不常见

2XX(成功状态码) 请求正常处理完毕

3XX(重定向状态码) 需要进行附加操作已完成请求

4XX(客户端错误状态码) 服务器无法请求类别

5XX(服务器错误状态码) 服务器处理请求出错

HTTP状态码一共有60多种，经常使用的大概14种

2XX 成功：(1)200 ok 客户端请求在服务器端正常处理

(2)204 No Content 请求成功但没有资源可以返回

(3)206 Partial Content 表示客户进行了范围请求，而服务器成功执行了这部分的GET请求 首部字段中有content-rage意思是客户端进行了范围请求

3XX 重定向：(1)301 Moved Permanently 永久性重定向

(2)302 Found 临时性重定向

(3)303 See Other 请求的对应资源存在着另一个URI，应使用GET方法定向获取请求的资源

(4)304 Not Modified 资源已找到，但是未符合要求

(5)307 Temporary Redirect 临时重定向

4XX 客户端错误：(1)400 Bad Request请求报文存在语法错误

(2)401 Unauthorized 发送的请求需要通过HTTP认证的认证信息。如果之前请求过一次，则表示用户认证失败

(3)403 Forbidden 不允许访问该资源

(4)404 Bad Request 服务器上没有该资源

5XX 服务器错误：(1)500 Internal Server Error服务器端在执行请求时发生错误

(2)503 Service Unavailable 服务器暂时处于超负载或正在停机维护，无法处理请求

注：不少返回的状态码响应是错误的，但是用户可能察觉不到这点，计算机也会出错，不一定绝对可靠

1. HTTP报文及其结构

HTTP的通信过程包括从客户端发往服务器端的请求及从服务器端

HTTP报文：用于HTTP协议交互的信息被称为报文。请求端的HTTP报文叫请求报文，响应段的HTTP报文叫响应报文

请求报文结构：报文首部：请求行，请求.通用.实体首部字段，其他 空行 报文主体

响应报文结构：报文首部：状态行，响应.通用.实体首部字段，其他 空行 报文主体

请求行：包含请求的方法，请求URI和HTTP版本

状态行：包含表明相应结果的状态码，原因短语和HTTP版本

首部字段：包含表示请求和响应的各种条件和属性的各类首部，一般有四种首部，通用首部，请求首部，响应首部和实体首部

其他：可能包含HTTP的RFC里未定义的首部(Cookie等)

编码提升速率：

HTTP传输的时候，可以按照数据原貌进行传输，但也可以在传输过程中通过编码来提升传输速率。

编码传输的好处：可以有效处理大量访问请求

编码传输的缺点：消耗更多CPU等资源

我们要弄明白一个东西：

报文主体和实体主体的差异！

报文：HTTP通信中的基本单位

实体：作为请求或响应的有效载荷数据(补充项)被传输，其内容有实体首部和实体主体组成

通常报文主体等于实体主体，只有当传输中进行编码操作时，实体主体的内容发生变化，才导致它和报文主体发生差异

内容编码：压缩传输的内容编码，压缩方法通常会采用gzip,

compress(UNIX系统的)，identity(不进行编码)

分块传输编码(性能优化)：

在HTTP传输过程中，请求的编码实体资源尚未全部传输完成，浏览器无法显示页面，所以大量传输数据时候，通常把数据分割成多块，能够让浏览器逐步显示页面。所以，这种把实体主体分块的功能称为分块传输编码。

多部分对象集合：

多部分对象集合的用途：发送多种数据(包括文本，视频，图片等)

邮件中采用了MIME机制(多用途因特网邮件扩展机制)，可以处理文本，图片，视频等不同类型的数据

HTTP协议支持这样做，在首部中添加Content-type

范围请求：

获取部分内容的范围请求：

作用：从之前下载中断处恢复下载

需要在首部前设置一个字段：

Content-Range: bytes=5001-10000

针对范围请求，返回的状态码为：

206 Partial Content

如果无法响应范围请求，则会返回状态码：

200 OK和完整的实体内容

内容协商：

当浏览器的默认语言为英语或者中文，访问相同的URI的web页面，则会是对应的英语版或者中文的web页面，这种机制叫内容协商

内容协商机制会以相应资源的语言，字符集，编码方式等作为判断标准

首部字段：Accept Accept-Chaeset Accept-Encoding Accept-

Language Content-Language

内容协商技术类型：

1. 服务器驱动协商：由服务器进行内容协商，以请求字段为参考，在服务器端自动处理
2. 客户端驱动协商：由客户端进行内容协商的方式，自己选
3. 透明协商：服务器驱动协商+客户端驱动协商
4. HTTP首部字段

HTTP首部字段是构成HTTP报文的要素之一，在客户端与服务器端之间以HTTP协议进行通信的过程中，无论是请求还是响应，都会用到首部字段，它能起到额外传递重要信息的作用，使用首部字段是为了为了浏览器和服务器提供报文主体大小.所使用语言，认证信息等内容，首部字段可以附加特别多的信息，而且不仅仅是有的头部信息，也可以是其他没有规定的

首部字段的结构：首部字段名：字段值

比如：Content-Type：text/html 首部字段可以有多个

首部字段的类型：一共被分为四种类型

1. 通用首部字段：请求报文和响应报文都会使用的首部
2. 请求首部字段：从客户端发送请求报文时使用的首部，补充了请求的附加内容，客户端信息，相应内容相关优先级等信息
3. 响应首部字段：从服务器端向客户端返回响应报文时使用的首部，补充响应的附加内容，也会要求客户端附加额外的内容信息
4. 实体首部字段：针对请求.响应报文实体部分使用的首部，补充了资源内容更新时间等与实体有关的信息

HTTP/1.1 首部字段一览表(文件夹内图片)

为Cookie服务的首部字段

Set-Cookie：开始状态管理所使用的Cookie信息(说明)，首部字段类型(响应首部字段)

Set-Cookie：yingchao = 3; expires = tue, 05 Jul 2011 7:26:31

GMT; =>path/; domain= .hackr.jp;Secure;

字段属性：NAME = VALUE 赋予Cookie的属性名和属性值

expires=DATA Cookie有效期(如果不指明，默认为浏览器关闭前为止)

path=PATH 将服务器上的文件目录作为Cookie的适用对象

domain=域名 作为Cookie适用对象的域名(若不指定则默认为创建服务器的域名)

Secure 仅在HTTPS安全通信是才会发送Cookie

HttpOnly 加以限制，适用Cookie无法访问js

第一部分 ajax

Asynchronous JavaScript + xml

异步的

包含的技术：(1)使用html，css来表示

1. 使用dom模型来交互和动态显示
2. 使用XMLHttpRequest来和服务器进行异步通信
3. 使用JavaScript来绑定和调用

优点：(1)不必手动刷新，在页面内与服务器通信，自动更新新的网页信息(定时器)

1. 使用异步的形式与服务器通信，不需要打断用户的操作
2. 减轻服务器负担
3. 不需要插件和任何小程序，应用性非常广泛

缺点：(1)不支持浏览器后退机制

1. 安全问题，跨站点脚本攻击，sqi注入攻击
2. 对搜索引擎的支持比较弱(seo)
3. 不支持手持设备，如pad，手机
4. 违反了url定位资源的初衷，找到的不一定就是地址的资源

ajax原理和XMLHttpRequest对象

原理：通过XMLHttpRequest对象来向服务器发送异步请求，从服务器获得数据，然后用JavaScript来操作DOM来更新页面

XMLHttpRequest对象 核心

属性：

onreadystatechange：每次状态改变时所触发的事件

responseText：从服务器进程返回数据的字符串形式

responseXML：从服务器进程返回的DOM兼容的文档数据对象

Status：从服务器返回的数字代码，比如404,200等等

Status Text：伴随状态码的字符串信息

readyState：状态值

0:未初始化(对象已经建立但未初始化，尚未调用open方法)

1:初始化(对象已经初始化，尚未调用send方法)

open有三个参数(“get/post”,”duyi.txt”,”true”),从前台送请求到后台,文件,ture是否异步

2:发送数据(send方法已经调用)方法为get没有参数，post要传数据xhr.send(data)

3:数据传输中

4:完成

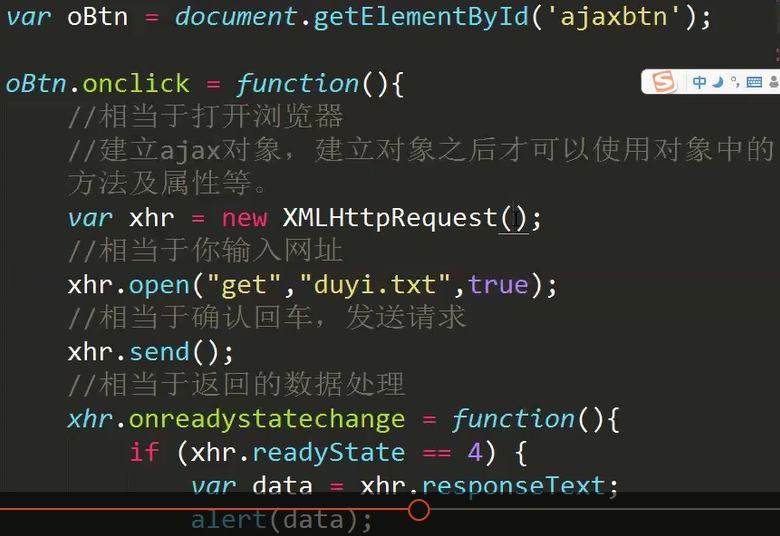
访问服务器环境127.0.0.1或localhost localhost/duyi/1

.html

在IE6及以下没有XMLHttpRequest，它有一个ActiveXObject插件中有一个对象Microsoft.XMLHttp，兼容写法：

if(window.XMLHtmlRequest){var xhr = XMLHttpRequest();}

else{var xhr = new ActiveObject(Microsoft.XMLHttp)}



xhr.readyState == 4 && xhr.status == 200

JSON和JSONP

JSON：是一种轻量级的数据交换格式，是完全独立于语言的文本格式（支持很多种的语言)，JSON没有变量的概念，是个对象，

xml阅读性好，JSON方便取值

1. 简单值：string，number，boolean，null，不支持undefined
2. 复杂值：对象，对象的属性值的类型是string，number，对象和数组

数组，数组值的类型为string，number，对象和数组

最简单的JSON数据形式就是简单值，eg：5,”a”JSON字符串必须用双引号，JSON对象无需声明变量，属性必须加双引号,最后也不用加分号

{

“name”:”Nicholas”,

“age”:29

}

Var person = {

Name:’hgf’

}；

数组同样无需声明变量和末尾分号

JSON序列化与解析方法：

可以把JSON数据结构解析为有用的JavaScript对象。

(1)JSON.stringify(obj)把JavaScript对象序列化为JSON字符串

(2)JSON.parse(obj)把JSON字符串解析为原生JavaScript对象

注意：同一个对象转换为JSON字符串再转回来跟原先对象一点关系都没有

JSON.parse(xhr.responseText)得到传过来的数据

详解stringify方法：三个参数，二三参数选填

对象，过滤器(可以是数组也可以是函数)，缩进选项(四空格，八空格缩进，最多十空格)

JSON.stringify(book,[“name”]/函数,4)，只取出了属性名为name的属性

详解parse方法：两个参数，二选填

JSON字符串，还原函数function(key,value)value

JSONP:JSON with padding

交换数据的方式

同源策略：是一种约定，它是浏览器最核心最基本的安全功能，如果缺少了同源策略，则浏览器的正常功能都会受到影响，可以说web构建在同源策略之上

同源是指域名，协议，端口相同，在同源的情况下请求资源才会请求成功

同源策略虽然保证了一定的安全性，但也阻碍了不同域间的相互协助，去其他域请求数据的操作

用JSONP解决跨域问题

利用script标签没有跨域限制的漏洞来达到与第三方通信的目的，用script标签指向第三方API地址，在并提供一个回调函数来接收数据(函数名可以约定，或者通过地址参数传递)第三方产生的响应应为JSON数据的包装(故称为JSONP)，形如callback

({“name”:”a”,”sex”:”male”})，这样浏览器就会调用callback函数，并传递解析后JSON对象作为参数

?wd=d&cb=hello，百度搜索框就是添加一个事件，筛选输入字符为wd的参数为关键词，改变cb，cb返回的jsonp也不一样，callback函数以字符串形式传入

豆瓣开发者，电影专栏