**HTTP协议总结**

**HTTP协议（超文本传输协议）**

http 是一个基于请求与响应模式的，无状态的，应用层的协议，该协议基于TCP链接（三次握手），HTTP 1.1版本中给出一种持续链接的机制，绝大多数的Web开发都是构建在HTTP协议之上的。

URL 是一种特殊类型的URI(统一资源标识符),包含用于查找某个资源的足够信息。

HTTP URL 格式如下：

http://host[":"port][abs\_path]

* 1
* 1
* host: 表示合法的internet主机域名和IP地址。
* port指端口号，为null则表示使用缺省的端口80（默认值）。
* abs\_path：指定请求资源的URI。如果URL中没有给出abs\_path，那么当它作为请求URI时，必须以“/”的形式给出，通常这个工作浏览器自动帮我们完成。

当我们输入www.hao123.com

http://www.hao123.com/

* 1
* 1

hao123 表示域名，输入该域名会在DNS服务器中查找该域名对应的ip。

该域名使用的是80端口，可以不需要输入。

最后因为没有输入资源地址，浏览器会默认在其后加入/作为结束。

**HTTP 协议的特点**

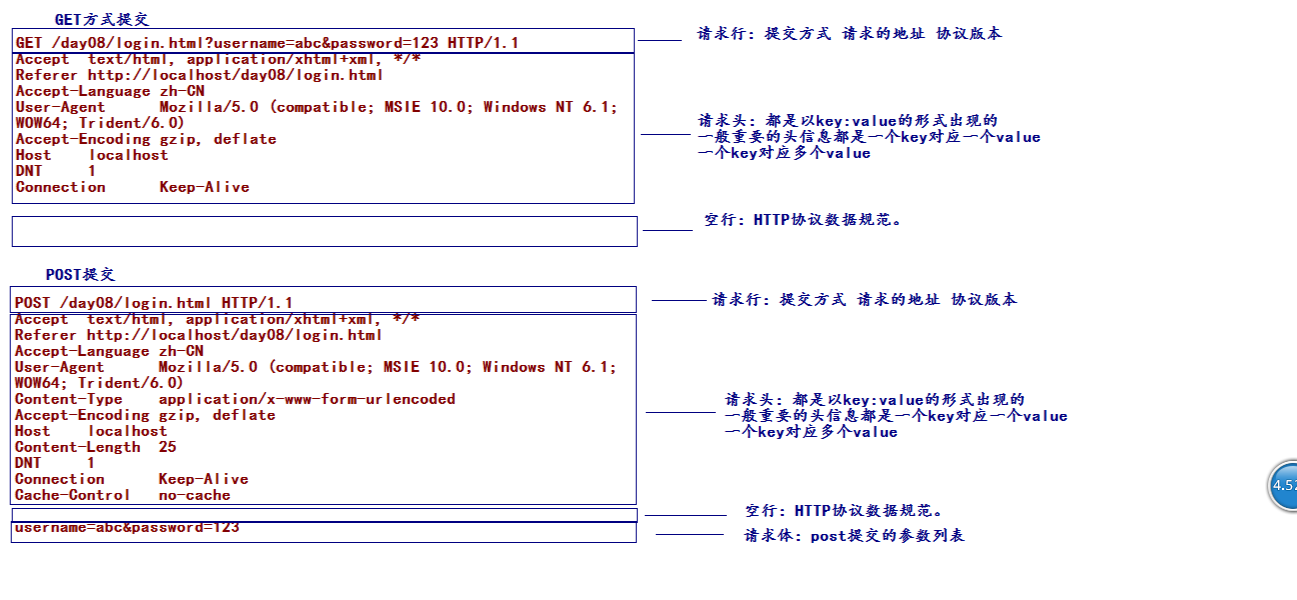
* 支持客户端/服务端模式。及浏览器可以和我们的服务器通过HTTP协议进行交互。
* 简单快速：客户向服务端请求服务时，只需传送请求方法和路径。请求方法常有的有GET,HEAD,POST。每种方法规定了客户与服务器联系的类型。由于HTTP协议的简单，使得HTTP协议HTTP服务器规模小，因而通信速度快。
* 灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。
* 无状态：对于事务的处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则必须重新传送。另一方面，在不需要前面的信息，处理速度较快。
* 无连接： 在HTTP /1.0 时，请求响应之后就断开。HTTP/1.1 ,请求之后不会立即断开，而是链接一段时间之后，如果没有后续请求，才会断开。

**HTTP 协议版本**

HTTP协议总共分为两个版本，分别是HTTP/1.0和HTTP/1.1.

* HTTP/1.0
  + 链接后，只能请求一个web资源
  + 链接后，只能做出一次响应和请求，响应完成之后服务器会立即断开。
* HTTP/1.1
  + 链接后，可以请求多个web资源
  + 链接后，发送请求，服务器做出响应，链接不会立即断开。再次发送请求，直接有一段时间没操作，自动断开。

**HTTP 请求**



由图中可以看出，HTTP 请求分为三部分，请求行，请求头，请求体。

* 请求行 分为三部分，
  + 请求方式：
    - 所有请求方式：POST、GET、HEAD、OPTIONS、DELETE、TRACE、PUT、CONNECT
    - 常用的请求方式有POST,GET。
    - POST和GET的区别：   
      * POST 将参数封装到请求体中，安全级别高，支持大数据。
      * GET 将参数直接显示到地址栏中，安全级别低，不支持大数据。
  + 请求地址： 请求的资源。
  + 协议版本：HTTP/1.1
* 请求头

Accept: text/html,image/\*

Accept-Charset: ISO-8859-1

Accept-Encoding: gzip

Accept-Language:zh-cn

Host: www.itcast.com:80

If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT

Referer: http://www.itcast.com/index.jsp

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)

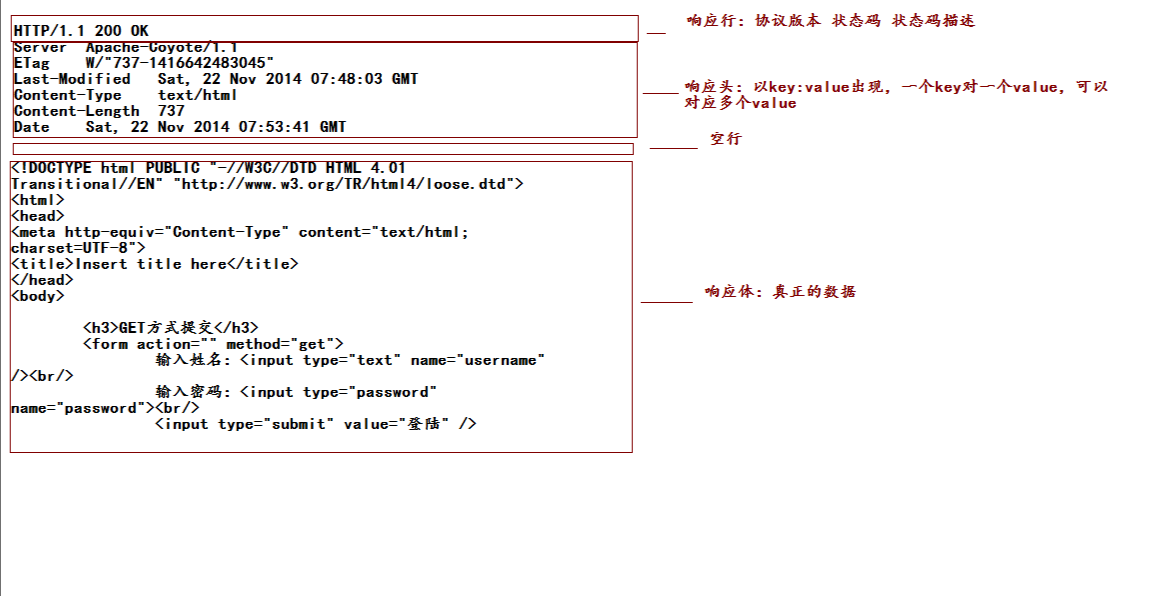
Connection: close/Keep-Alive

Date: Tue, 11 Jul 2000 18:23:51 GMT

请求体： GET为null.Post 封装参数列表。

可以看到请求头重有一些字段，该字段都是以键-值对的形式出现。某些字段是一键对多值。我们可以通过设置该字段的一些值以达到一些特殊的效果。

**HTTP 响应**



响应和请求很类似，分为响应行，响应头，响应体。

其中在响应行中有200 ，该字段标识请求响应的结果。

状态代码有三位数字组成，第一个数字定义了响应的类别，且有五种可能取值：

* 1xx：指示信息–表示请求已接收，继续处理
* 2xx：成功–表示请求已被成功接收、理解、接受
* 3xx：重定向–要完成请求必须进行更进一步的操作
* 4xx：客户端错误–请求有语法错误或请求无法实现
* 5xx：服务器端错误–服务器未能实现合法的请求

常见的响应码如下：

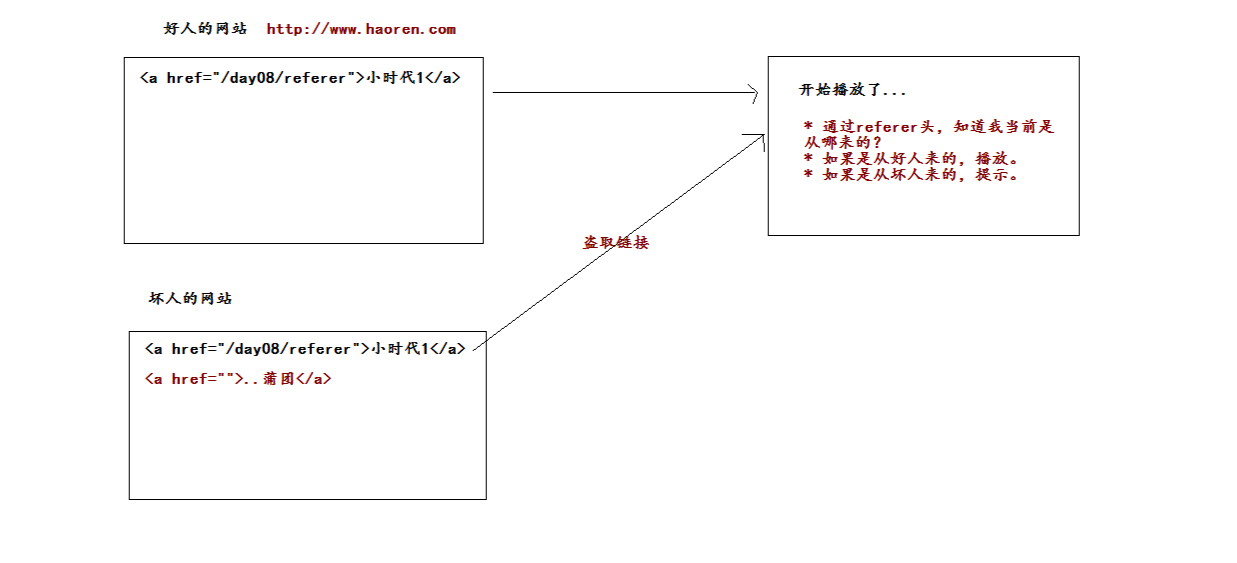
* 200 客户端请求成功。
* 400 客户端请求有语法错误，不能服务器所理解。
* 401 请求未经授权
* 404 请求资源不存在
* 500 服务器发生不可预知的错误
* 503 服务器当前不能处理客户端的请求，一段时间后可能恢复正常。

在相应头中，类似请求头一样，也是存在着一个个键值对。可以配合请求头做一些事情。

**HTTP 的应用**

防盗链

在一些网站中，可以获取别人网站的链接加入到自己的网站中，导致用户从其他网站上浏览到了本站的信息。

   
如图，有两个网站，分别为好人和坏人的网站，其中坏人盗取了好人的网站链接。即盗链。

而在请求中存在字段referer头信息，可以通过referer判断链接是否正确。

而在JavaWeb中，通过Servlet的request可以获取头信息。

//通过referer 和自己的网址进行对比

String referer = request.getHeader("Referer");

获取浏览器信息

通过请求头的user-agent获取浏览器信息。

// 防止中文乱码

response.setContentType("text/html;charset=UTF-8");

// 获取浏览器信息

String s = request.getHeader("user-agent");

response.getWriter().write(s);

Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36

获取到浏览器信息，系统信息。 通过该字段可以区分访问时来自pc端还是移动端。

页面的定时刷新

通过响应头字段refresh实现页面定时刷新。

response.setContentType("text/html;charset=UTF-8");

response.getWriter().write("访问到了...");

// 页面5秒会跳转

response.setHeader("refresh", "5;url=http://www.baidu.com");

页面会在5秒之后跳转到百度。

实现重定向

转发和重定向的区别

* 转发：找班长借钱，他自己找富班长借钱。（两个页面拼接到一起，后台悄悄处理）
* 重定向：找班长借钱，发送一次请求，回了我没钱，返回状态码302，给副班长地址，再去找富班长借钱，又发送了一次。

重定向的实现

response.setContentType("text/html;charset=UTF-8");

// response.getWriter().write("向班长借钱...");

// 我没钱

response.setStatus(302);

// 告诉我富班长的地址

response.setHeader("location", "副班长地址");