http by taitanxiami

http://www.cnblogs.com/rayray/p/3729533.html (课件相符)
1. 什么是HTTP协议 超文本传输协议是一种详细规定了浏览器和万维网服务器之间互相通信的规则，通过因特网传送万维网文档的数据传送协议。

HTTP(超文本传输协议)是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。它于1990年提出，经过几年的使用与发展，得到不断地完善和扩展。目前在WWW中使用的是HTTP/1.0的第六版，HTTP/1.1的规范化工作正在进行之中，而且HTTP-NG(Next Generation of HTTP)的建议已经提出。

1.1建立连接方面 HTTP/1.0 每次请求都需要建立新的TCP连接，连接不能复用。HTTP/1.1 新的请求可以在上次请求建立的TCP连接之上发送，连接可以复用。优点是减少重复进行TCP三次握手的开销，提高效率。

注意：在同一个TCP连接中，新的请求需要等上次请求收到响应后，才能发送。

HTTP协议的主要特点可概括如下：

1.支持`客户/服务器模式`。
2.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。
3.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。
4.`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。
5.`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。（这次的请求和上次的请求没有关系，后引入缓存机制cookie）

URL

URL(Uniform Resource Locator) 地址用于描述一个网络上的资源,
基本格式如下 : schema://host[:port#]/path/.../[;url-params][?query-string][#anchor]

HTTP协议详解之请求篇

> `http请求由三部分组成，分别是：请求行、消息报头（http header）、请求正文（body）`

1、请求行以一个方法符号开头，以空格分开，后面跟着请求的URI和协议的版本，格式如下：**Method Request-URI HTTP-Version CRLF** 其中 Method表示请求方法；Request-URI是一个统一资源标识符；HTTP-Version表示请求的HTTP协议版本；CRLF表示回车和换行（除了作为结尾的CRLF外，不允许出现单独的CR或LF字符）。
`HTTP请求消息实例：`

```
GET /hello.htm HTTP/1.1
Accept: */*     浏览器可接受的MIME类型;
Accept-Language: zh-cn
Accept-Encoding: gzip, deflate
If-Modified-Since: Wed, 17 Oct 2007 02:15:55 GMT
If-None-Match: W/"158-1192587355000"
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 192.168.2.162:8080
Connection: Keep-Alive
```

请求方法（所有方法全为大写）有多种，各个方法的解释如下:

```
GET     请求获取Request-URI所标识的资源
POST    在Request-URI所标识的资源后附加新的数据
HEAD    请求获取由Request-URI所标识的资源的响应消息报头
PUT     请求服务器存储一个资源，并用Request-URI作为其标识
DELETE  请求服务器删除Request-URI所标识的资源
TRACE   请求服务器回送收到的请求信息，主要用于测试或诊断
CONNECT 保留将来使用
OPTIONS 请求查询服务器的性能，或者查询与资源相关的选项和需求
```

POST方法要求被请求服务器接受附在请求后面的数据，常用于提交表单。

```
eg: POST /reg.jsp HTTP/ (CRLF)
Accept:image/gif,image/x-xbit,... (CRLF)
...
HOST:www.guet.edu.cn (CRLF)
Content-Length:22 (CRLF)
Connection:Keep-Alive (CRLF)
Cache-Control:no-cache (CRLF)
(CRLF)          //该CRLF表示消息报头已经结束，在此之前为消息报头
user=jeffrey&pwd=1234  //此行以下为提交的数据
```

HEAD方法与GET方法几乎是一样的，对于HEAD请求的回应部分来说，它的HTTP头部中包含的信息与通过GET请求所得到的信息是相同的。利用这个方法，不必传输整个资源内容，就可以得到Request-URI所标识的资源的信息。该方法常用于测试超链接的有效性，是否可以访问，以及最近是否更新。 2、请求报头后述 3、请求正文(略)

HTTP协议详解之响应篇

在接收和解释请求消息后，服务器返回一个HTTP响应消息。

```
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
```

1、状态行格式如下：
HTTP-Version Status-Code Reason-Phrase
其中，HTTP-Version表示服务器HTTP协议的版本；Status-Code表示服务器发回的响应状态代码；Reason-Phrase表示状态代码的文本描述。

HTTP响应消息实例如下所示：

```
HTTP/1.1 200 OK
ETag: W/"158-1192590101000"
Last-Modified: Wed, 17 Oct 2007 03:01:41 GMT
Content-Type: text/html
Content-Length: 158
Date: Wed, 17 Oct 2007 03:01:59 GMT
Server: Apache-Coyote/1.1
```

5.常用的请求方式

常用的请求方式是GET和POST.

GET方式：是以实体的方式得到由请求URI所指定资源的信息，如果请求URI只是一个数据产生过程，那么最终要在响应实体中返回的是处理过程的结果所指向的资源，而不是处理过程的描述。

POST方式：用来向目的服务器发出请求，要求它接受被附在请求后的实体，并把它当作请求队列中请求URI所指定资源的**附加新子项**，Post被设计成用统一的方法实现下列功能：

从上面描述可以看出，Get是向服务器发索取数据的一种请求；而Post是向服务器提交数据的一种请求，要提交的数据位于信息头后面的实体中。

GET与POST方法有以下区别：

（1） 在客户端Get方式在通过URL提交数据，数据在URL中可以看到；POST方式，数据放置在HTML HEADER内提交。
1. GET提交的数据会放在URL之后，以?分割URL和传输数据，参数之间以&相连，如EditPosts.aspx?name=test1&id=123456. POST方法是把提交的数据放在HTTP包的Body中.
2. GET提交的数据大小有限制（因为浏览器对URL的长度有限制），而POST方法提交的数据没有限制.
3. GET方式需要使用Request.QueryString来取得变量的值，而POST方式通过Request.Form来获取变量的值。
4. GET方式提交数据，会带来安全问题，比如一个登录页面，通过GET方式提交数据时，用户名和密码将出现在URL上，如果页面可以被缓存或者其他人可以访问这台机器，就可以从历史记录获得该用户的账号和密码.

6.请求头

HTTP最常见的请求头如下：

Accept：浏览器可接受的MIME类型；

Accept-Charset：浏览器可接受的字符集；

`Accept-Encoding`：浏览器能够进行解码的数据编码方式，比如gzip。Servlet能够向支持gzip的浏览器返回经gzip编码的HTML页面。许多情形下这可以减少5到10倍的下载时间；

`Accept-Language`：浏览器所希望的语言种类，当服务器能够提供一种以上的语言版本时要用到；

`Authorization`：授权信息，通常出现在对服务器发送的WWW-Authenticate头的应答中；

Connection：表示是否需要持久连接。如果Servlet看到这里的值为“Keep-Alive”，或者看到请求使用的是HTTP 1.1（HTTP 1.1默认进行持久连接），它就可以利用持久连接的优点，当页面包含多个元素时（例如Applet，图片），显著地减少下载所需要的时间。要实现这一点，Servlet需要在应答中发送一个Content-Length头，最简单的实现方法是：先把内容写入ByteArrayOutputStream，然后在正式写出内容之前计算它的大小；

Content-Length：表示请求消息正文的长度；

Cookie：这是最重要的请求头信息之一；

From：请求发送者的email地址，由一些特殊的Web客户程序使用，浏览器不会用到它；

Host：初始URL中的主机和端口；

If-Modified-Since：只有当所请求的内容在指定的日期之后又经过修改才返回它，否则返回304“Not Modified”应答；

Pragma：指定“no-cache”值表示服务器必须返回一个刷新后的文档，即使它是代理服务器而且已经有了页面的本地拷贝；

Referer：包含一个URL，用户从该URL代表的页面出发访问当前请求的页面。

User-Agent：浏览器类型，如果Servlet返回的内容与浏览器类型有关则该值非常有用；

UA-Pixels，UA-Color，UA-OS，UA-CPU：由某些版本的IE浏览器所发送的非标准的请求头，表示屏幕大小、颜色深度、操作系统和CPU类型。

7.响应头

HTTP最常见的响应头如下所示：

Allow：服务器支持哪些请求方法（如GET、POST等）；
Content-Encoding：文档的编码（Encode）方法。只有在解码之后才可以得到Content-Type头指定的内容类型。利用gzip压缩文档能够显著地减少HTML文档的下载时间。Java的GZIPOutputStream可以很方便地进行gzip压缩，但只有Unix上的Netscape和Windows上的IE 4、IE 5才支持它。因此，Servlet应该通过查看Accept-Encoding头（即request.getHeader("Accept-Encoding")）检查浏览器是否支持gzip，为支持gzip的浏览器返回经gzip压缩的HTML页面，为其他浏览器返回普通页面；

Content-Length：表示内容长度。只有当浏览器使用持久HTTP连接时才需要这个数据。如果你想要利用持久连接的优势，可以把输出文档写入ByteArrayOutputStram，完成后查看其大小，然后把该值放入Content-Length头，最后通过byteArrayStream.writeTo(response.getOutputStream()发送内容； Content-Type： 表示后面的文档属于什么MIME类型。Servlet默认为text/plain，但通常需要显式地指定为text/html。由于经常要设置Content-Type，因此HttpServletResponse提供了一个专用的方法

setContentTyep。 可在web.xml文件中配置扩展名和MIME类型的对应关系；

Date：当前的GMT时间。你可以用setDateHeader来设置这个头以避免转换时间格式的麻烦；

Expires：指明应该在什么时候认为文档已经过期，从而不再缓存它。

Last-Modified：文档的最后改动时间。客户可以通过If-Modified-Since请求头提供一个日期，该请求将被视为一个条件GET，只有改动时间迟于指定时间的文档才会返回，否则返回一个304（Not Modified）状态。Last-Modified也可用setDateHeader方法来设置；

Location：表示客户应当到哪里去提取文档。Location通常不是直接设置的，而是通过HttpServletResponse的sendRedirect方法，该方法同时设置状态代码为302；

Refresh：表示浏览器应该在多少时间之后刷新文档，以秒计。除了刷新当前文档之外，你还可以通过setHeader("Refresh", "5; URL=http://host/path")让浏览器读取指定的页面。注意这种功能通常是通过设置HTML页面HEAD区的实现，这是因为，自动刷新或重定向对于那些不能使用CGI或Servlet的HTML编写者十分重要。但是，对于Servlet来说，直接设置Refresh头更加方便。注意Refresh的意义是“N秒之后刷新本页面或访问指定页面”，而不是“每隔N秒刷新本页面或访问指定页面”。因此，连续刷新要求每次都发送一个Refresh头，而发送204状态代码则可以阻止浏览器继续刷新，不管是使用Refresh头还是。注意Refresh头不属于HTTP 1.1正式规范的一部分，而是一个扩展，但Netscape和IE都支持它。

8.实体头

实体头用坐实体内容的元信息，描述了实体内容的属性，包括实体信息类型，长度，压缩方法，最后一次修改时间，数据有效性等。

`Allow`：GET,POST

`Content-Encoding`：文档的编码（Encode）方法，例如：gzip，见“2.5 响应头”；

`Content-Language`：内容的语言类型，例如：zh-cn；

`Content-Length`：表示内容长度，eg：80，可参考“2.5响应头”；

`Content-Location`：表示客户应当到哪里去提取文档，例如：http://www.dfdf.org/dfdf.html，可参考“2.5响应头”；

`Content-MD5`：MD5 实体的一种MD5摘要，用作校验和。发送方和接受方都计算MD5摘要，接受方将其计算的值与此头标中传递的值进行比较。Eg1：Content-MD5: 。Eg2：dfdfdfdfdfdfdff==；
`Content-Range`：随部分实体一同发送；标明被插入字节的低位与高位字节偏移，也标明此实体的总长度。Eg1：Content-Range: 1001-2000/5000，eg2：bytes 2543-4532/7898
Content-Type：标明发送或者接收的实体的MIME类型。Eg：text/html; charset=GB2312 主类型/子类型；

Expires：为0证明不缓存；

Lat-Modified：WEB 服务器认为对象的最后修改时间，比如文件的最后修改时间，动态页面的最后产生时间等等。例如：Last-Modified：Tue, 06 May 2008 02:42:43 GMT.

容易犯的误区：

HTTP协议是无状态的和Connection: keep-alive的区别

1、HTTP是一个无状态的面向连接的协议，无状态不代表HTTP不能保持TCP连接，更不能代表HTTP使用的是UDP协议（无连接）
2、从HTTP/1.1起，默认都开启了Keep-Alive，保持连接特性，简单地说，当一个网页打开完成后，客户端和服务器之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接
3、Keep-Alive不会永久保持连接，它有一个保持时间，可以在不同的服务器软件（如Apache）中设定这个时间