HTTP(HyperText Transfer Protocol)是一套计算机通过网络进行通信的规则。计算机专家设计出HTTP，使HTTP客户（如Web浏览器）能够从HTTP服务器(Web服务 器)请求信息和服务HTTP是一种无状态的协议，无状态是指Web浏览器和Web服务器之间不需要建立持久的连接，这意味着当一个客户端向服务器端发出请求，然后Web服务器返回响应(response)，连接就被关闭了，在服务器端不保留连接的有关信息.HTTP遵循 请求(Request)/应答(Response)模型。Web浏览器向Web服务器发送请求，Web服务器处理请求并返回适当的应答。所有HTTP连接 都被构造成一套请求和应答。

HTTP 使用内容类型，是指Web服务器向Web浏览器返回的文件都有与之相关的类型。所有这些类型在MIME　Internet邮件协议上模型化，即Web服务 器告诉Web浏览器该文件所具有的种类，是HTML文档、GIF格式图像、声音文件还是独立的应用程序。

HTTP通信机制是在一次完整的HTTP通信过程中，Web浏览器与Web服务器之间将完成下列7个步骤：

（1）  建立TCP连接

在HTTP 工作开始之前，Web浏览器首先要通过网络与Web服务器建立连接，该连接是通过TCP来完成的，该协议与IP协议共同构建Internet，即著名的 TCP/IP协议族，因此Internet又被称作是TCP/IP网络。HTTP是比TCP更高层次的应用层协议，根据规则，只有低层协议建立之后才能， 才能进行更层协议的连接，因此，首先要建立TCP连接，一般TCP连接的端口号是80

（2）    Web浏览器向Web服务器发送请求命令

一旦建立了TCP连接，Web浏览器就会向Web服务器发送请求命令

例如：GET/sample/hello.jsp HTTP/1.1

（3）    Web浏览器发送请求头信息

浏览器发送其请求命令之后，还要以头信息的形式向Web服务器发送一些别的信息，之后浏览器发送了一空白行来通知服务器，它已经结束了该头信息的发送。

（4）    Web服务器应答

客户机向服务器发出请求后，服务器会客户机回送应答，

HTTP/1.1 200 OK

应答的第一部分是协议的版本号和应答状态码

（5）    Web服务器发送应答头信息

正如客户端会随同请求发送关于自身的信息一样，服务器也会随同应答向用户发送关于它自己的数据及被请求的文档。

（6）    Web服务器向浏览器发送数据

Web服务器向浏览器发送头信息后，它会发送一个空白行来表示头信息的发送到此为结束，接着，它就以Content-Type应答头信息所描述的格式发送用户所请求的实际数据

（7）    Web服务器关闭TCP连接

一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接，然后如果浏览器或者服务器在其头信息加入了这行代码

Connection:keep-alive

TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽。

HTTP请求格式

当浏览器向Web服务器发出请求时，它向服务器传递了一个数据块，也就是请求信息，HTTP请求信息由3部分组成：

l   请求方法URI协议/版本

l   请求头(Request Header)

l   请求正文

下面是一个HTTP请求的例子：

GET/sample.jspHTTP/1.1

Accept:image/gif.image/jpeg,\*/\*

Accept-Language:zh-cn

Connection:Keep-Alive

Host:localhost

User-Agent:Mozila/4.0(compatible;MSIE5.01;Window NT5.0)

Accept-Encoding:gzip,deflate

username=jinqiao&password=1234

（1）       请求方法URI协议/版本

请求的第一行是“方法URL议/版本”：GET/sample.jsp HTTP/1.1

以上代码中“GET”代表请求方法，“/sample.jsp”表示URI，“HTTP/1.1代表协议和协议的版本。

根据HTTP标准，HTTP请求可以使用多种请求方法。例如：HTTP1.1支持7种请求方法：GET、POST、HEAD、OPTIONS、PUT、DELETE和TARCE。在Internet应用中，最常用的方法是GET和POST。

URL完整地指定了要访问的网络资源，通常只要给出相对于服务器的根目录的相对目录即可，因此总是以“/”开头，最后，协议版本声明了通信过程中使用HTTP的版本。

（2）　请求头(Request Header)

请求头包含许多有关的客户端环境和请求正文的有用信息。例如，请求头可以声明浏览器所用的语言，请求正文的长度等。

Accept:image/gif.image/jpeg.\*/\*

Accept-Language:zh-cn

Connection:Keep-Alive

Host:localhost

User-Agent:Mozila/4.0(compatible:MSIE5.01:Windows NT5.0)

Accept-Encoding:gzip,deflate.

（3）　请求正文

请求头和请求正文之间是一个空行，这个行非常重要，它表示请求头已经结束，接下来的是请求正文。请求正文中可以包含客户提交的查询字符串信息：

username=jinqiao&password=1234

在以上的例子的HTTP请求中，请求的正文只有一行内容。当然，在实际应用中，HTTP请求正文可以包含更多的内容。

HTTP请求方法我这里只讨论GET方法与POST方法

l         GET方法

GET方法是默认的HTTP请求方法，我们日常用GET方法来提交表单数据，然而用GET方法提交的表单数据只经过了简单的编码，同时它将作为URL的一部分向Web服务器发送，因此，如果使用GET方法来提交表单数据就存在着安全隐患上。例如

[Http://127.0.0.1/login.jsp?Name=zhangshi&Age=30&Submit=%cc%E+%BD%BB](http://127.0.0.1/login.jsp?Name=zhangshi&Age=30&Submit=%cc%25E+%BD%BB)

从上面的URL请求中，很容易就可以辩认出表单提交的内容。（？之后的内容）另外由于GET方法提交的数据是作为URL请求的一部分所以提交的数据量不能太大

l         POST方法

POST 方法是GET方法的一个替代方法，它主要是向Web服务器提交表单数据，尤其是大批量的数据。POST方法克服了GET方法的一些缺点。通过POST方法 提交表单数据时，数据不是作为URL请求的一部分而是作为标准数据传送给Web服务器，这就克服了GET方法中的信息无法保密和数据量太小的缺点。因此， 出于安全的考虑以及对用户隐私的尊重，通常表单提交时采用POST方法。

　　从编程的角度来讲，如果用户通过GET方法提交数据，则数据存放在QUERY＿STRING环境变量中，而POST方法提交的数据则可以从标准输入流中获取。

HTTP应答与HTTP请求相似，HTTP响应也由3个部分构成，分别是：

l　协议状态版本代码描述

l　响应头(Response Header)

l　响应正文

下面是一个HTTP响应的例子：

HTTP/1.1 200 OK

Server:Apache Tomcat/5.0.12

Date:Mon,6Oct2003 13:23:42 GMT

Content-Length:112

<html>  
<head>

<title>HTTP响应示例<title>

</head>

<body>

Hello HTTP!

</body>

</html>  
协议状态代码描述HTTP响应的第一行类似于HTTP请求的第一行，它表示通信所用的协议是HTTP1.1服务器已经成功的处理了客户端发出的请求（200表示成功）:

HTTP/1.1 200 OK  
响应头(Response Header)响应头也和请求头一样包含许多有用的信息，例如服务器类型、日期时间、内容类型和长度等：

　　 Server:Apache Tomcat/5.0.12

Date:Mon,6Oct2003 13:13:33 GMT

Content-Type:text/html

Last-Moified:Mon,6 Oct 2003 13:23:42 GMT

Content-Length:112

 响应正文响应正文就是服务器返回的HTML页面：

　　<html>  
<head>

<title>HTTP响应示例<title>

</head>

<body>

Hello HTTP!

</body>

</html>

响应头和正文之间也必须用空行分隔。

l        HTTP应答码

　　　HTTP应答码也称为状态码，它反映了Web服务器处理HTTP请求状态。HTTP应答码由3位数字构成，其中首位数字定义了应答码的类型：

　　　1XX－信息类(Information),表示收到Web浏览器请求，正在进一步的处理中

　　　2XX－成功类（Successful）,表示用户请求被正确接收，理解和处理例如：200 OK

      3XX-重定向类(Redirection),表示请求没有成功，客户必须采取进一步的动作。

      4XX-客户端错误(Client Error)，表示客户端提交的请求有错误 例如：404 NOT

                                    Found，意味着请求中所引用的文档不存在。

      5XX-服务器错误(Server Error)表示服务器不能完成对请求的处理：如 500

      对于我们Web开发人员来说掌握HTTP应答码有助于提高Web应用程序调试的效率和准确性。

安全连接

Web 应用最常见的用途之一是电子商务，可以利用Web服务器端程序使人们能够网络购物，需要指出一点是，缺省情况下，通过Internet发送信息是不安全 的，如果某人碰巧截获了你发给朋友的一则消息，他就能打开它，假想在里面有你的信用卡号码，这会有多么糟糕，幸运的是，很多Web服务器以及Web浏览器 都有创立安全连接的能力，这样它们就可以安全的通信了。

通过Internet 提供安全连接最常见的标准是安全套接层(Secure Sockets layer,SSl)协议。SSL协议是一个应用层协议(和HTTP一样)，用于安全方式在Web上交换数据，SSL使用公开密钥编码系统。从本质讲，这 意味着业务中每一方都拥有一个公开的和一个私有的密钥。当一方使用另一方公开密钥进行编码时，只有拥有匹配密钥的人才能对其解码。简单来讲，公开密钥编码 提供了一种用于在两方之间交换数据的安全方法，SSL连接建立之后，客户和服务器都交换公开密钥，并在进行业务联系之前进行验证，一旦双方的密钥都通过验 证，就可以安全地交换数据。

* GET  
  通过**请求**URI得到资源
* POST,  
  用于添加新的内容
* PUT  
  用于修改某个内容
* DELETE,  
  删除某个内容
* CONNECT,  
  用于代理进行传输，如使用SSL
* OPTIONS  
  询问可以执行哪些**方法**
* PATCH,  
  部分文档更改
* PROPFIND, (wedav)  
  查看属性
* PROPPATCH, (wedav)  
  设置属性
* MKCOL, (wedav)  
  创建集合（文件夹）
* COPY, (wedav)  
  拷贝
* MOVE, (wedav)  
  移动
* LOCK, (wedav)  
  加锁
* UNLOCK (wedav)  
  解锁
* TRACE  
  用于远程诊断服务器
* HEAD  
  类似于GET, 但是不返回body信息，用于检查对象是否存在，以及得到对象的元数据