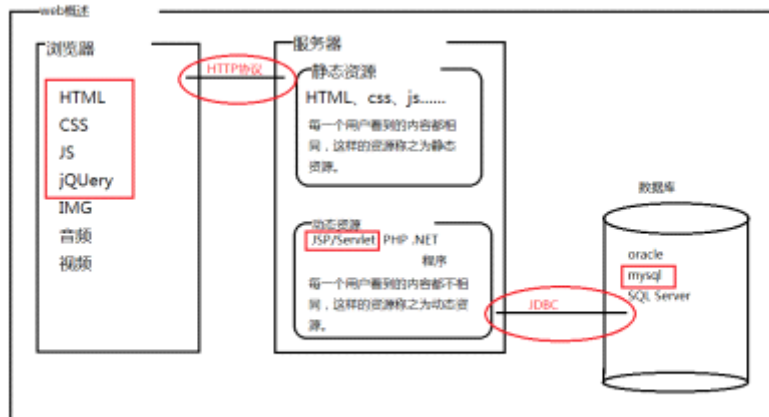


## 1. web开发概述



### 静态资源:

HTML、CSS、JS、jQuery

--- 用户看到的内容都相同，这些资源称之为静态资源。

### 动态资源:

Servlet/JSP

--- 用户看到的内容不相同的资源称之为动态资源。

--- 由程序处理的资源。

## 1. Web是什么？

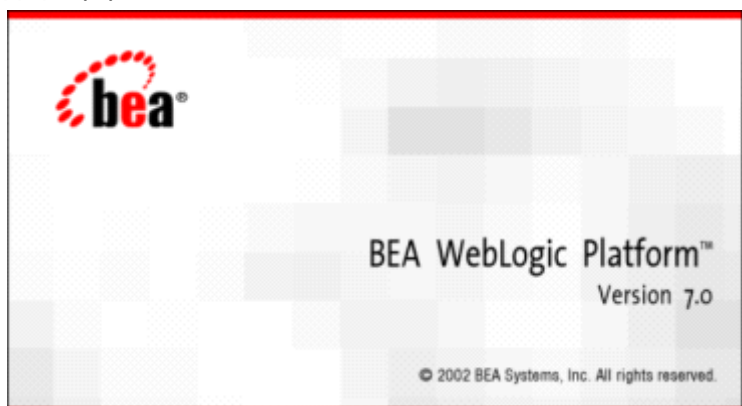
web即为网页的意思，表示在Internet主机上可供其他用户访问的资源。

## 2. web服务器

服务器软件运行在物理服务器上。在物理服务器上安装一个服务器软件即可运行服务器软件并进行相关的web开发。

## 3. 常见WEB服务器

WebLogic是BEA公司的产品，是目前应用最广泛的Web服务器，支持J2EE规范，而且不断的完善以适应新的开发要求，启动界面如图



另一个常用的Web服务器是IBM公司的WebSphere，支持J2EE规范，启动界面如图



在小型的应用系统或者有特殊需要的系统中，可以使用一个 免费的Web服务器：Tomcat，该服务器支持全部JSP以及Servlet规范，启动界面如图



## 1. Tomcat服务器简介



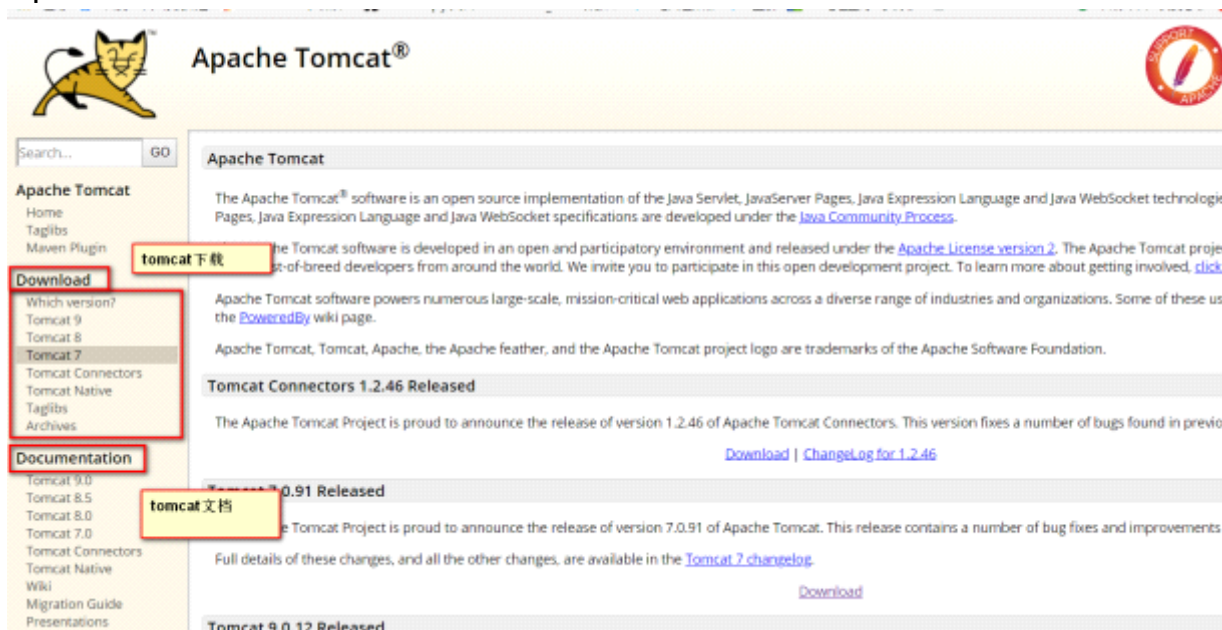
tomcat是一台小型服务器。支持JSP、servlet技术。  
在tomcat面对高并发的时候是否能正常运行？

建立tomcat集群可以解决高并发的这个问题。

## 2. Tomcat服务器安装和配置

### a. Tomcat 的下载与安装

- 下载地址：<http://tomcat.apache.org/>
- 获取Tomcat安装程序包
  - tar.gz文件是Linux操作系统下的安装版本
  - exe文件是Windows系统下的安装版本
  - zip文件是Windows系统下的压缩版本



Apache Tomcat®

Search... GO

Apache Tomcat

- Home
- Taglibs
- Maven Plugin

Download

- Which version?
- Tomcat 9
- Tomcat 8
- Tomcat 7
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation

- Tomcat 9.0
- Tomcat 8.5
- Tomcat 8.0
- Tomcat 7.0
- Tomcat Connectors
- Tomcat Native
- Wiki
- Migration Guide
- Presentations

Tomcat 7 Software Downloads

Welcome to the Apache Tomcat® 7.x software download page. This page provides download links for obtaining the latest version of Tomcat 7.0.x software releases.

Quick Navigation

[KEYS](#) [7.0.91](#) [Browse](#) | [Archives](#)

Release Integrity

You **must** verify the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against keys of Tomcat's Release Managers. We also provide SHA-512 checksums for every release file. After you download the file, you should calculate a checksum the same as ours.

Mirrors

You are currently using <http://mirrors.shu.edu.cn/apache/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are of the mirrors list that should be available.

Other mirrors:  [Change](#)

7.0.91

Please see the [README](#) file for packaging information. It explains what every distribution contains.

7.0.91

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
  - [32-bit Windows.zip \(pgp, sha512\)](#)
  - [64-bit Windows.zip \(pgp, sha512\)](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
  - [tar.gz \(pgp, sha512\)](#)
- Deployer:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
- Extras:
  - [JMX Remote.jar \(pgp, sha512\)](#)
  - [Web services.jar \(pgp, sha512\)](#)
  - [JULI adapters.jar \(pgp, sha512\)](#)
  - [JULI log4j.jar \(pgp, sha512\)](#)
- Embedded:
  - [tar.gz \(pgp, sha512\)](#)
  - [zip \(pgp, sha512\)](#)

Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

## ■ 安装Tomcat

### □ 注意事项（常见问题）：

- ◆ 安装目录不能包含中文和空格。
- ◆ JAVA\_HOME环境变量指定Tomcat运行时所要用的jdk所在的位置，注意，配到目录就行了，不用指定到bin。
- ◆ 端口占用问题：netstat -ano命令查看端口占用信息。
- ◆ Catalina\_Home环境变量：startup.bat启动哪个tomcat由此环境变量指定，如果不配置则启动当前tomcat，推荐不要配置此环境变量。

### □ 安装完成后：

双击 tomcat/bin 目录下的 startup.bat 文件

在浏览器端输入 <http://localhost:8080/> 显示如下界面代表安装成功



## b. Tomcat的目录结构

bin --- 存储和tomcat启动关闭相关的脚本文件

conf --- 存储和tomcat配置相关的一个目录。

lib --- 存储tomcat启动和运行时必要的jar包。

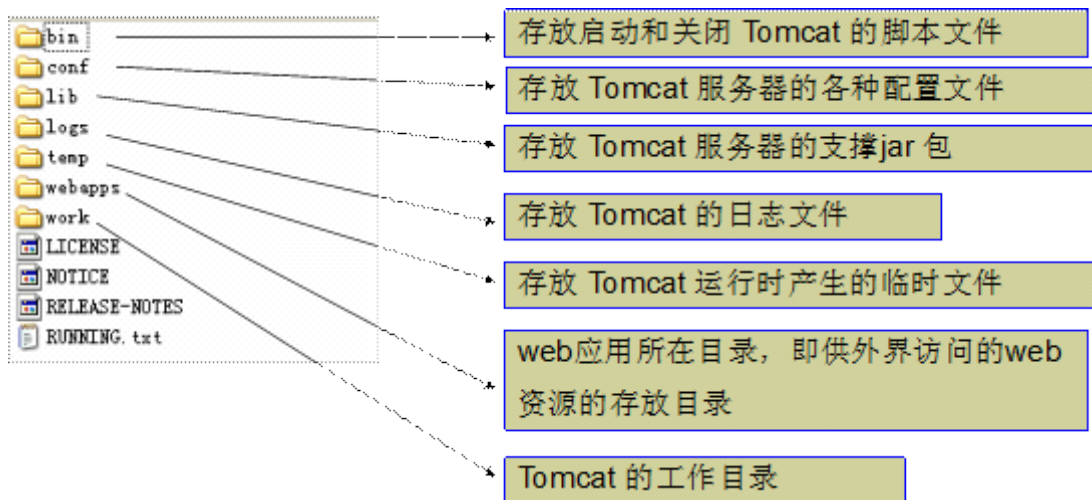
logs --- 存储tomcat启动和运行过程中的日志信息文件，通过查阅这些文件可以调试tomcat。

temp --- 存储tomcat临时文件的目录。

webapps --- 存储tomcat的web资源的一个目录。

work --- 存储tomcat的在运行之后产生的页面信息文件的目录。

- 修改server.xml文件中的第71行，将8080修改为80端口，这个端口即为一个默认端口，可以在访问的时候不书写。（在浏览器中不书写端口号即表明访问默认80端口）



### 1. 虚拟主机

在一个tomcat服务器中，tomcat可以被认为是一台真是的主机，在一台主机中可以配置多个网站，这些网站在访问者看来是运行在各自独立主机之上，但是开发者知晓这两个网站上运行在同一个主机当中。那么就可以说这个网站运行在真实主机身上的一台虚拟主机当中。一个网站就是一台虚拟主机。

### 2. web应用的概念

一个web应用主要是存储web的一个目录。由于在虚拟主机当中，web资源无法直接存储在虚拟主机当中，就需要在虚拟主机中建立一个目录，利用这个目录存储对应功能的web资源，像这样的目录称之为web应用程序，简称web应用。一般情况下根据功能划分web应用，一个功能相关的web资源放在同一个web应用的目录下。

### 3. web应用

- 在web应用中应该存储对应功能的全部web资源。
- web应用创建成功之后是希望被用户访问到的，访问一个web应用，需要在地址栏中输入一个web应用的虚拟地址，这个虚拟地址应该对应服务器上真实的web应用名称。
- 创建web应用的映射关系：
  - web应用的映射：虚拟地址与真实路径的关系。url --- 真实文件的路径

### 4. web应用的配置：

#### 第一种web应用的配置：

在server.xml中的<Host>标签内部，添加如下内容：

```
<Context path="/news1" docBase="D:/news" />
```

其中path="虚拟web应用的路径" docBase="真实web应用的路径"。配置之后会在服务中生成映射真实路径映射到了虚拟路径身上，在地址栏中输入/news1即可访问D盘下的news web应用。

#### 第二种web应用的配置：

在[tomcat]/conf/Catalina/localhost目录中添加一个.xml文件，这个文件的名称就是web应用的虚拟路径，在文件中添加内容如下：

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<Context docBase="D:/email" />
```

其中docBase="真实web应用的目录"。

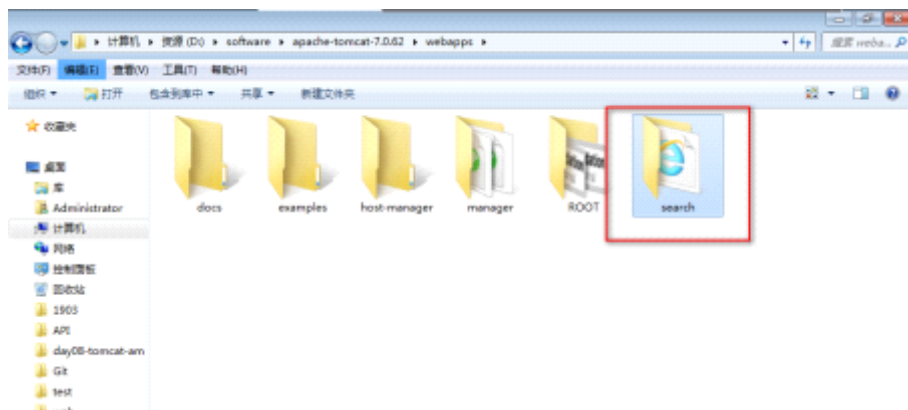
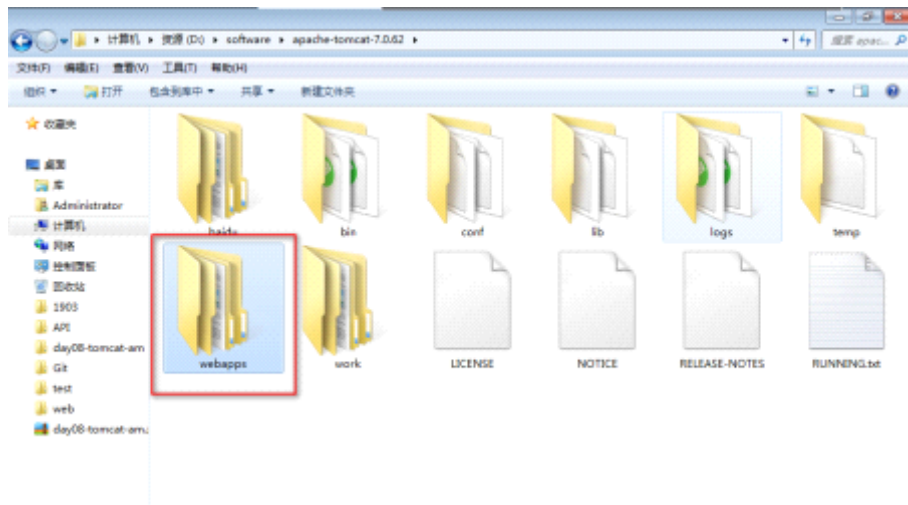
**注意：**第二种配置方式中如果要求虚拟路径有多级，例如：/email/aaa，在Windows中文件命名不允许使用"/"所以可以用"#"来代替，"#"表示将虚拟路径分级的意思，在地址栏中输入对应分级后路径即可。

#### 第三种web应用的配置：

在localhost虚拟主机管理的webapps目录中添加一个目录，这个目录的名称就是web应用的虚拟路径，同时这个目录也是真实的web应用。所以只需添加此处一个web应用即完成全部配置。

配置方式如下：





缺省web应用的配置:

一个路径被配置为缺省的路径之后, 在地址栏中就不需要书写当前web应用的虚拟路径。

i. 第一种web应用配置缺省:

将path中包含的路径完全删除, 只留下一对引号, 这就表示当前web应用的虚拟路径为缺省web应用的路径。

```
<Context path="" docBase="D:/news" />
```

ii. 第二种web应用配置缺省:

将Catalina/[虚拟主机名称]/email.xml文件名称修改为ROOT.xml文件, 这是当前文件的名称就是一个缺省web应用的虚拟路径。

iii. 第三种web应用配置缺省:

将webapps目录中的search目录名称修改为ROOT, 则当前目录的名称就会变成一个web应用的缺省路径。

**注意:** 缺省web应用的虚拟路径有优先级:

server.xml > Catalina/[虚拟主机名称]/ROOT.xml文件 > webapps目录中的ROOT目录

## 5. web应用目录结构

web应用的目录结构:

web应用可以只有它本身这一个目录, 也可以在目录中添加一个子级目录WEB-INF, 如果有这个目录则一定要保证这个目录结构的完整。

web应用

|  
|---1.html 1.css 此处可以存放静态资源，可以被用户直接访问。

|----WEB-INF

|---静态资源 此处可以存放静态资源，但是无法被用户直接访问。（并不是必须存在的资源。）

|--- lib 存储当前web应用运行时所需jar包。

|--- classes 存储当前web应用中所有的动态资源的目录。

|--- web.xml 此文件必须存在于完整的web应用目录结构中。

servlet映射的配置/监听器/过滤器/缺省主页

a. 在web.xml中配置一个缺省主页：

如果一个页面作为缺省主页使用，则在地址栏中无需书写页面名称即可作为默认主页访问。

i. 复制[tomcat]/conf/web.xml，复制内容如下：

并且修改下方welcome-file-list标签中index.html为1.html页面。  
表示将1.html页面作为一个缺省主页使用。

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd
  version="3.0">

  <welcome-file-list>
    <welcome-file>1.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

ii. 在地址栏中直接书写localhost即可访问到缺省主页(前提是已经配置了缺省web应用)

iii. 在配置缺省主页时可以配置多个，但是只取第一个存在的页面。后边的页面都不会生效。

b. web.xml文件作用：

i. 配置servlet映射的路径

ii. 配置过滤器

iii. 配置监听器

iv. 配置缺省主页。

## 6. 虚拟主机的配置

a. 在server.xml文件中复制一份<Host>标签，并且将这个标签粘贴到</Host>结束标签之后。

代码如下：



```

<Valve className="org.apache.catalina.valves.
AccessLogValve" directory="logs"
        prefix="localhost_access_log." suffix=".tx
        pattern="%h %l %u %t &quot;%r&quot; %s %b"
--><Context path="/" docBase="D:/news" />-->
</Host>

<Host name="www.baidu.com" appBase="baidu">

</Host>

```

重新启动tomcat服务器，使配置生效，重新启动时会自动创建baidu目录。

## b. 修改hosts文件

hosts文件路径：

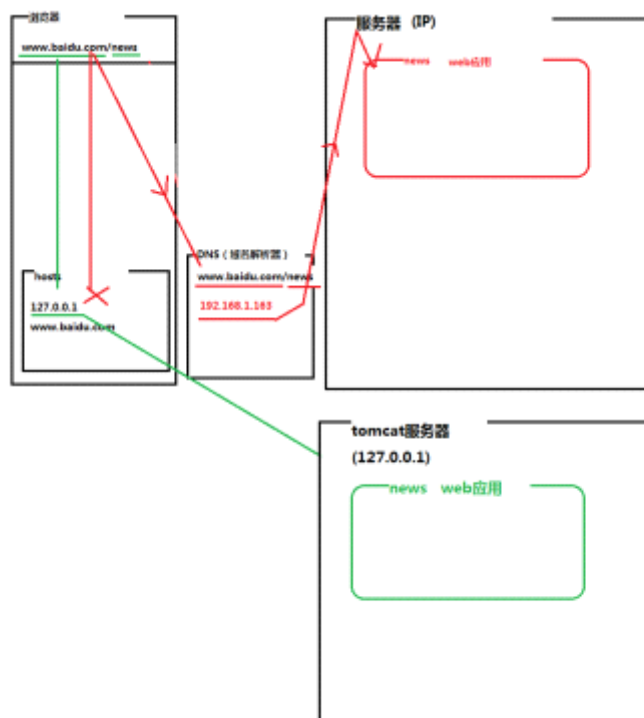
C:\Windows\System32\drivers\etc\hosts

在其中添加代码如下：

127.0.0.1 [www.baidu.com](http://www.baidu.com)

此句表示将www.baidu.com指向当前主机，也就是访问自己主机上的www.baidu.com, 而不是访问互联网中的百度网。

hosts与DNS解析器：



## • <Host>有这些属性：

属性	描述	必需？
className	默认为org.apache.catalina.core.StandardHost	否
appBase	web应用程序文件存放的位置，相对路径为CATALINA_HOME	是
autoDeploy	设为true，则web.xml发生变化时，tomcat自动重新部署程序。实现这个功能必需允许后台处理	否
name	virtual host的名称	是
backgroundProcessingDelay	跟Engine中的backgroundProcessingDelay类似	否
deployOnStartup	若为true，则当这个Engine启动时，tomcat将自动部署这个host，默认为true	否

deployXML	这个属性的目的是为了提高tomcat的安全性，控制web应用程序是否能使用META-INF/context.xml。如果设为false，则各应用程序只能访问\$CATALINA_HOME/conf/<engine>/<host>/<app>.xml。默认值为True。	否
errorReportValveClasses	定义host使用的error-reporting Valve，默认值为org.apache.catalina.valves.ErrorReportValve	否
unpackWARs	tomcat在webapps文件夹中发现war文件时，是否自动将其解压	否
workdir	tomcat使用这个目录来放工作着的servlet和jsp（以servlet形式），这里的servlet都是编译好的class文件。默认为\$CATALINA_HOME/work	

#### c. 缺省虚拟主机

如果用户在地址栏中输入一个ip地址，则当前ip地址与服务器上的所有虚拟主机翻译成的ip地址都匹配，这时tomcat服务器就会自动选择server.xml中<Engine>，身上指定的缺省虚拟主机的地址来提供服务。

代码如下：

```
<Engine name="Catalina" defaultHost="localhost">
```

#### 7. 杂项：

##### a. 打war包

在web应用的目录下打开一个cmd创建，书写如下内容：

```
jar -cvf ROOT.war *
```

##### b. 压缩方式打war包：

直接进入web应用内部，选中全部文件添加一个压缩包，名称中修改后缀为.zip进行压缩，压缩完成之后，将后缀再修改为.war即可。

#### 8. 作业：

- 创建一个www.google.com虚拟主机，并且将虚拟主机配置成缺省虚拟主机，在其中配置两个web应用一个名称为news一个名称为email。并且修改其中任意一个web应用为缺省web应用，为缺省web应用配置一个缺省主页。

## 1. 什么是HTTP协议？

- a. 在浏览器端连接上服务器端之后，才会进行数据的交互，HTTP协议主要规定了浏览器和服务端是通过何种形式进行数据通信。HTTP就是数据通信的一种形式。
- b. HTTP HyperText Transfer protocol（超文本传输协议）。
- c. OSI网络七层模型：  
应用层(HTTP) -- 表示层 -- 会话层 -- 传输层 -- 网络层 -- 数据链路层 -- 物理层

## 2. HTTP协议

基于请求响应模型。

一次请求对应一次响应。

请求只能有浏览器发出。服务器根据请求做出响应。

## 3. HTTP/1.0和HTTP/1.1

在建立连接之后，HTTP/1.0协议会在发送一次请求接受一次响应之后断开连接。

在建立连接之后，HTTP/1.1协议可以发送多次请求，针对每次请求做出响应，在这个过程中连接一直是建立状态，在浏览器经过一段时间不再发送请求之后，连接才断开。

- 利用telnet演示HTTP1.0和HTTP1.1的区别
  - 首先准备一个web应用，并且tomcat是启动状态。
  - 打开cmd窗口，输入telnet localhost 端口号
  - 按ctrl+]
  - 回车，进入输入界面（输入界面不允许删除字符，因为每输入一个字符就将这个字符以流的形式发送给服务器，无法修改。）
  - 输入指令 GET /web应用/资源名 HTTP/1.1
  - 回车，输入Host:localhost:端口号
- 使用HTTP1.1协议和1.0协议分别测试，发现1.1执行完一次请求后等待下一次请求，1.0则在一次请求后断开。
- **telnet测试注意事项**：每次telnet测试完成后要退出重进，必要时需要重启tomcat。
- a. 问题：
  - i. 如果在页面中有三张图片，请问访问这个页面发送几次请求？建立的几次连接呢？

答：在页面中有三张图片所以需要三次请求，还有一个页面也需要发送一次请求，所以一共发送4次请求。

由于使用的是HTTP/1.1协议所以建立一次连接。

## 4. HTTP请求

- 客户端连上服务器后，向服务器请求某个web资源，称之为客户端向服务器发送了一个HTTP请求。一个完整的HTTP请求包括如下内容：

- 一个请求行、若干请求头、一个空行、以及实体内容，如下所示：

- 举例：

```
GET /books/java.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Connection: Keep-Alive
Host: localhost
Referer: http://localhost/linka.asp
User-Agent: Mozilla/4.0
Accept-Encoding: gzip, deflate
```

← 请求行

请求行用于描述客户端的请求方式、请求的资源名称，以及使用的HTTP协议版本号

← 多个请求头

消息头用于描述客户端请求哪台主机，以及客户端的一些环境信息等

← 一个空行

← 实体内容

## 5. HTTP请求的细节——请求行

- 请求行中的GET称之为请求方式，请求方式有：
- POST、GET、HEAD、OPTIONS、DELETE、TRACE、PUT
- 常用的有：GET、POST
- 用户如没有设置，默认情况下浏览器向服务器发送的都是get请求，例如在浏览器直接输地址访问，点超链接访问等都是get，用户如想把请求方式改为post，可通过更改表单的提交方式实现。
- 不管POST或GET，都用于向服务器请求某个WEB资源，这两种方式的区别主要表现在数据传递上：
  - 如请求方式为GET方式，则可以在请求的URL地址后以?的形式带上交给服务器的数据，多个数据之间以&进行分隔，例如：
    - GET /mail/1.html?name=abc&password=xyz HTTP/1.1
  - GET方式的特点：在URL地址后附带的参数是有限制的，其数据容量通常不能超过1K。
  - 如请求方式为POST方式，则可以在请求的实体内容中向服务器发送数据，Post方式的特点：传送的数据量无限制。

## 6. HTTP请求的细节——消息头

- 用于HTTP请求中的常用头
- Accept: text/html,image/\* 通知浏览器可以接受什么数据 现在表示接收文本数据,任意格式的图片数据
- Accept-Charset: ISO-8859-1
- Accept-Encoding: gzip,compress 发送数据的压缩格式
- Accept-Language: en-us,zh-cn 语言
- Host: [www.it315.org:80](http://www.it315.org:80) 请求的地址,可以通过观察Host头来确定访问对应地址服务器上的哪个虚拟主机。

- If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT 与响应头last modified呼应
- Referer: <http://www.it315.org/index.jsp> 防盗链
- User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
- Cookie
- Connection: close/Keep-Alive
- Date: Tue, 11 Jul 2000 18:23:51 GMT

## 7. HTTP响应

- 一个HTTP响应代表服务器向客户端回送的数据，它包括：
- 一个状态行、若干响应头(消息头)、一个空行、以及实体内容。

### ● 举例：

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 13 Jul 2000 05:45:53 GMT
Content-Length: 2291
Content-Type: text/html
Cache-control: private

<HTML>
<BODY>
...

```

← 状态行

状态行用于描述服务器对请求的处理结果。

← 多个响应头

消息头用于描述服务器的基本信息，以及数据的描述，服务器通过这些数据的描述信息，可以通知客户端如何处理等一会儿它回送的数据。

← 一个空行

← 实体内容

代表服务器向客户端回送的数据

## 8. HTTP响应的细节——状态行

### 状态行

格式：HTTP版本号 状态码 原因叙述<CRLF>

举例：HTTP/1.1 200 OK

状态码用于表示服务器对请求的处理结果，它是一个三位的十进制数。响应状态码分为5类，如下所示：

状态码	含义
100 ~ 199	表示成功接收请求，要求客户端继续提交下一次请求才能完成整个处理过程
200 ~ 299	表示成功接收请求并已完成整个处理过程，常用200
300 ~ 399	为完成请求，客户需进一步细化请求。例如，请求的资源已经移动一个新地址，常用302、307和304
400 ~ 499	客户端的请求有错误，常用404
500 ~ 599	服务器端出现错误，常用 500

200 页面加载成功

304 307使用缓存

302 +location 请求重定向

404 资源找不到

500 502 503 504 505

## 9. HTTP响应细节——常用响应头

- HTTP请求中的常用响应头
- Location: <http://www.it315.org/index.jsp> 配合302实现请求重定向
- Server:apache tomcat 服务器类型
- Content-Encoding: gzip 服务器发送数据的压缩格式
- Content-Length: 80 发送数据的长度
- Content-Language: zh-cn 发送数据的语言环境
- Content-Type: text/html; charset=GB2312 可接受数据格式和语言
- Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 与请求头的if modified since头呼应
- Refresh: 1;url=http://www.it315.org 定时跳转
- Content-Disposition: attachment;filename=aaa.zip
- Transfer-Encoding: chunked
- Set-Cookie:SS=Q0=5Lb\_nQ; path=/search
- ETag: W/"83794-1208174400000"
- Expires: -1 通知浏览器是否缓存当前资源，如果这个头的值是一个以毫秒为单位的值就是通知浏览器缓存资源到指定的时间点，如果值是0或-1则是通知浏览器禁止缓存。
- Cache-Control: no-cache 通知浏览器是否缓存的头
- Pragma: no-cache 通知浏览器是否缓存的头
- Connection: close/Keep-Alive
- Date: Tue, 11 Jul 2000 18:23:51 GMT