# web服务器与HTTP协议

## Web服务器

* WEB，在英语中web即表示网页的意思，它用于表示Internet主机上供外界访问的资源。
* Internet上供外界访问的Web资源分为：
  + 静态web资源（如html 页面）：指web页面中供人们浏览的数据始终是不变。
  + 动态web资源：指web页面中供人们浏览的数据是由程序产生的，不同时间点访问web页面看到的内容各不相同。
* 静态web资源开发技术
  + Html
* 常用动态web资源开发技术：
  + JSP/Servlet、ASP、PHP等 ruby python
  + 在Java中，动态web资源开发技术统称为Javaweb，我们课程的重点也是教大家如何使用Java技术开发动态的web资源，即动态web页面。

但是我们做java开发，不是做网页。

网络上的资源分为两种

早期:静态页面 html实现。 观看

现在:动态页面 php asp jsp 交互.

lamp =linux +apache+ mysql+php----->个人网关或小型企业首选

asp现在没人用，但是网络上遗留下来的比较多。miscrosoft的技术

.net技术。

jsp--->java去做网页所使用的技术。jsp本质上就是servlet

使用jsp开发成本高。

BS====>浏览器+服务器 只要有浏览器就可以

CS----->客户端+服务器. 必须的在客户端安装程序.

现在基本上开发的都是BS程序

BS怎样通信:

必须有请求有响应。

有一次请求就应该具有一次响应，它们是成对出现的。

### 服务器介绍

大型服务器：websphere(IBM),weblogic(Oracle)　J2EE容器　－

支持EJB (EnterPrice Java Bean (企业级的javabean)) – Spring

weblogic BEA公司产品，被Oracle收购，全面支持JavaEE规范，收费软件，企业中非常主流的服务器 -------- 网络上文档非常全面

WebSphere 文档非常少，IBM公司产品，价格昂贵，全面支持JavaEE 规范

Tomcat- apache，开源的。Servlet容器。

tomcat 开源小型web服务器 ，完全免费，主要用于中小型web项目，只支持Servlet和JSP 等少量javaee规范 ，Apache公司jakarta 一个子项目

Jboss – hibernate公司开发。不是开源免费。J2EE容器

### Tomcat安装

注意路径中不要包含空格与中文。

* 安装步骤

1、tomcat.apache.org 下载tomcat安装程序

tomcat6 安装程序 ---- zip免安装版

2、解压tomcat

3、配置环境变量 JAVA\_HOME 指向JDK安装目录 D:\Program Files\Java\jdk1.6.0\_21

\*CATALINA\_HOME指定tomcat安装目录

4、双击tomcat/bin/startup.bat

5、在浏览器中 输入 localhost:8080 访问tomcat主页了

* 注意问题：

启动黑色不能关闭

1、CATALINA\_HOME 指定tomcat安装位置 --- 可以不配置

2、JAVA\_HOME 指定JDK安装目录，不要配置bin目录，不要在结尾加;

3、端口被占用

启动cmd

netstat -ano 查看占用端口进程pid

任务管理器 查看---选择列 显示pid -- 根据pid结束进程

\* 有些进程无法关系（系统服务 --- 必须结束服务） win7 自带 World wide web publish IIS服务 默认占用端口80

\* xp 安装apache服务器后，会占用80 端口 ，关闭apache服务

通过运行 services.msc 打开服务窗口 关闭相应服务

### tomcatc目录结构

-----bin 它里面装入的是可执行的命令 如 startup.bat

-----conf 它里面是一个相关的配置文件，我们可以在里面进行例如端口，用户信息的配置

<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

-----lib tomcat类库。

-----logs tomcat 日志文件

-----temp 临时文件

-----webapps 它里面放的是的 web site(web项目)

-----work 存放的是页面(例如 jsp)转换成的.class文件。

## 创建网站，将网站发布到tomcat服务器上

### 创建网站根目录

在根目录下 创建静态web资源和动态web资源

Web site

---- \*.html \*.css \*.js 图片 音频 视频 、\*.jsp

---- WEB-INF目录 存放java程序和配置文件

--- classes 存放.class文件

--- lib 存放.jar 文件

--- web.xml 网站核心配置文件

\*\*\* 如果静态网站可以不存在 WEB-INF目录的 ，WEB-INF目录，客户端无法直接访问（在服务器内存通过程序访问）

### 将网站发布到tomcat -----------虚拟目录映射

虚似目录的映射方式有三种

1.在开发中应用的比较多 直接在webapps下创建一个自己的web site就可以.

步骤 1.在webapps下创建一个myweb目录

2.在myweb下创建WEB-INF目录，在这个目录下创建web.xml

3.将web.xml文件中的xml声明与根元素声明在其它的web site中copy过来。

4.在myweb下创建一个index.html文件

5.启动tomcat

6.在浏览器中输入 http://localhost/myweb/index.html

以下两种方式，可以将web site不放置在tomcat/webapps下，可以任意放置

2.在server.xml文件中进行配置

<Context path="/abc" docBase="C:\myweb1"/>

</Host>在Host结束前配置

path：它是一个虚拟路径，是我们在浏览器中输入的路径

docBase：它是我们web sit的真实路径

http://localhost/abc/index.html

3.不在server.xml文件中配置

而是直接创建一个abc.xml文件

在这个xml文件中写

<Context path="" docBase="C:\myweb1"/>

将这个文件放入conf\Catalina\localhost

http://localhost/abc/index.html

## 生成war文件

war文件是web项目的压缩文件。

要想生成，先将要压缩的内容压缩成zip文件，

然后将后缀改成war就可以，

war文件可以直接在服务器上访问。

关于tomcat-manager

可以在conf/tomcat-users.xml中进行用户信息添加

<role rolename="manager"/>

<user username="xxx" password="xx" roles="manager"/>

这样就添加了一个用户

注意，用户权限要是比较大的话，会出现安全问题.

## 虚拟主机

做自己的一个http://www.baidu.com

1.访问一个网站的过程

http://www.baidu.com

http 协议

www 服务器

baidu.com 域名 IP

步骤

1.上网将baidu首页下载下来

2.做一个自己的web site 首页就是下载下来的页面。

别忘记创建WEB-INF在它下面的web.xml文件中

<welcome-file-list>

<welcome-file>index.html</welcome-file>

</welcome-file-list>

这句话的作用是默认访问页面是index.html

3.在tomcat中的conf文件夹下的server.xml中修改

<Host name="www.baidu.com" appBase="c:\baidu"

unpackWARs="true" autoDeploy="true"

xmlValidation="false" xmlNamespaceAware="false">

<Context path="" docBase="c:\baidu"/>

</Host>

4.在windows/system32/drivers/etc/hosts中添加

127.0.0.1 www.baidu.com

目的是当访问www.baidu.com时其实访问的是本机。

5.打开浏览器在地址栏中输入www.baidu.com

这时其时访问的是我们自己

web site中的页面。

## 使用myeclipse创建web project与tomcat集成

我们在myeclipse中创建web project有一个WebRoot目录。

但是我们发布到tomcat中没有这个，它其时就是我们工程的名称.

步骤

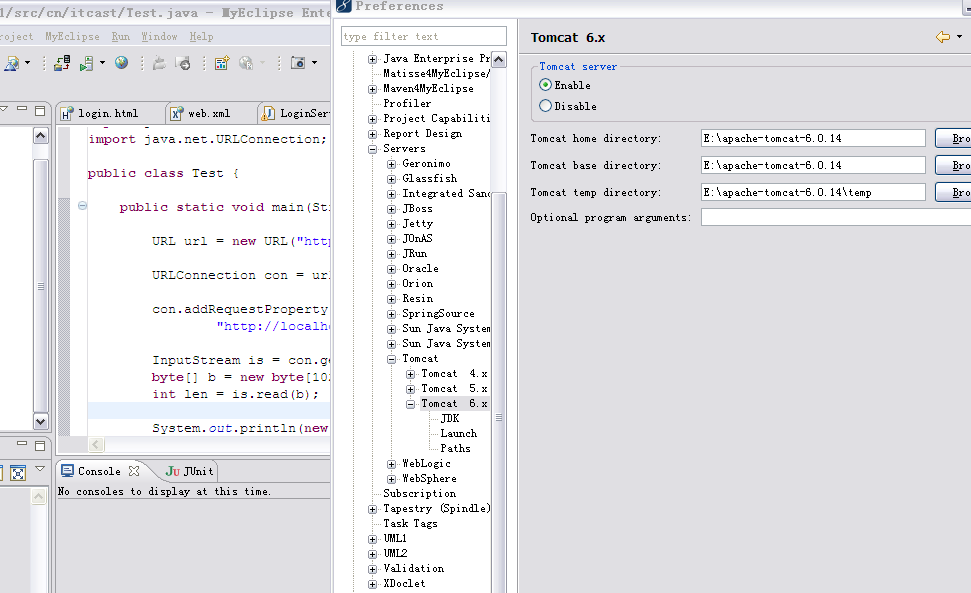
1.创建web工程

2.在eclipse中配置tomcat服务器

window/属性/myeclipse/service中配置自己的tomcat目录.

注意到tomcat根目录就可以了。不要到bin中。

如果不好使用，看一些jdk是否配置.



1. 将webproject部署到tomcat中

## HTTP协议

HTTP是hypertext transfer protocol（超文本传输协议）的简写，它是TCP/IP协议的一个应用层协议，用于定义WEB浏览器与WEB服务器之间交换数据的过程。

HTTP协议是学习JavaWEB开发的基石，不深入了解HTTP协议，就不能说掌握了WEB开发，更无法管理和维护一些复杂的WEB站点。

### 示例1

telnet怎样使用

1.telnet localhost 8080

2 ctrl+]

3.按回车

注意 在里面写错的内容不能修改

GET /index.html HTTP/1.1

host:localhost

4.要敲两次回车

HTTP/1.0版本只能保持一次会话

HTTP/1.1版本可能保持多次会话.

是根据telnet得到的响应信息

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

ETag: W/"7347-1184876416000"

Last-Modified: Thu, 19 Jul 2007 20:20:16 GMT

Content-Type: text/html

Content-Length: 7347

Date: Thu, 25 Apr 2013 08:06:53 GMT

Connection: close

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<title>Apache Tomcat</title>

<style type="text/css">

..........

### 示例2

是根据httpwatch得到的请求信息与响应信息

请求

GET / HTTP/1.1

Accept: application/x-shockwave-flash, image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, \*/\*

Accept-Language: zh-cn

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)

Accept-Encoding: gzip, deflate

Host: localhost

Connection: Keep-Alive

响应

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

ETag: W/"7347-1184876416000"

Last-Modified: Thu, 19 Jul 2007 20:20:16 GMT

Content-Type: text/html

Content-Length: 7347

Date: Thu, 25 Apr 2013 08:12:57 GMT

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang

### 请求信息详解

GET /books/java.html HTTP/1.1 ---------->请求行

Get是请求方式 /books/java.html 请求资源 HTTp/1.1协议版本

POST与GET的区别

1.什么样是GET 请求 1)直接在地址栏输入 2.超连接 <a></a> 3.form表单中method=get

什么样是POSt请求 form表单中method=POST

2.以get方式提交请求时，在请求行中会将提交信息直接带过去

格式 /day03\_1/login?username=tom&password=123

以post方式提交时，信息会在正文中。

POST /day03\_1/login HTTP/1.1

Accept: application/x-shockwave-flash, image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, \*/\*

Referer: http://localhost/day03\_1/login.html

Accept-Language: zh-cn

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)

Content-Type: application/x-www-form-urlencoded

Accept-Encoding: gzip, deflate

Host: localhost

Content-Length: 25

Connection: Keep-Alive

Cache-Control: no-cache

username=tom&password=123

3. get方式最多能提交1kb

post可以提交大数据，做上传时必须是post

Accept: \*/\* 允许访问mime类型,类型都在tomcat 的conf/web.xml文件中定义了。

这个需要知道，因为做下载时要知道mime类型

Accept-Language: en-us 客户端的语言

Connection: Keep-Alive 持续连接

Host: localhost 客户端访问资源

Referer: http://localhost/links.asp (重点) 防盗链。

User-Agent: Mozilla/4.0 得到浏览器版本 避免兼容问题

Accept-Charset: ISO-8859-1 客户端字符编码集

Accept-Encoding: gzip, deflate 浏览器支持的gzip是压缩编码.

If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT 与Last-MOdified一起可以控制缓存。

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

重点：

1.Referer ---防盗链

2.user--agent---可以获取浏览器的信息

3.if-modified--since-----与http相应中的一个响应头信息一起使用，可以完成缓存策略

示例1

*防盗链程序*

*referer.htm页面*

*<body>*

*<a href="referer">referer</a>*

*</body>*

*RefererServlet类似*

*public void doPost(HttpServletRequest request, HttpServletResponse response)*

*throws ServletException, IOException {*

*String msg = request.getHeader("Referer");*

*if (msg != null && "http://localhost/day03\_1/referer.html".equals(msg)) {*

*// 如果你是正常访问,我们给其一个友好信息*

*response.getWriter().write("hello");*

*} else {*

*// 如果是盗链过来的，对不。骂它一句*

*response.getWriter().write("fuck...");*

*}*

*}*

*怎样破解*

*URL url = new URL("http://localhost/day03\_1/referer"); //得到一个url*

*URLConnection con = url.openConnection(); //访问这个url，并获得连接对象*

*con.addRequestProperty("Referer",*

*"http://localhost/day03\_1/referer.html");*

*InputStream is = con.getInputStream(); // 读取服务器返回的信息.*

*byte[] b = new byte[1024];*

*int len = is.read(b);*

*System.out.println(new String(b, 0, len));*

### http协议响应

HTTP/1.1 200 OK 响应状态行

HTTP/1.1 200 OK

1xx 什么都没做直接返回

2xx 成功返回

3xx 做了一些事情，没有全部完成。

4xx 客户端错误

5xx 服务器错误

200 正确

302 重定向

304 页面没有改变

404 未找到页面

500 服务器出错.

Location: http://www.it315.org/index.jsp 响应路径(重点)+302

Server:apache tomcat

Content-Encoding: gzip 响应编码 gzip 压缩

Content-Length: 80 响应长度

Content-Language: zh-cn 响应语言

Content-Type: text/html; charset=GB2312 响应字符编码

Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 要与请求中的 If-Modified-Since处理缓存

Refresh: 1;url=http://www.it315.org 自动跳转

Content-Disposition: attachment; filename=aaa.zip (重要) 文件的下载

//下面三个是禁用浏览缓存

Expires: -1

Cache-Control: no-cache

Pragma: no-cache

Connection: close/Keep-Alive

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

重点

今天可以讲

Location: http://www.it315.org/index.jsp 响应路径(重点)+302

Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 要与请求中的 If-Modified-Since处理缓存

Refresh: 1;url=http://www.it315.org 自动跳转

我们在得到响应信息，经常得到的是压缩后的。

这种操作

1.服务器配置方式

tomcat配置实现压缩

80端口没有配置 00:00:00.000 0.228 7553 GET 200 text/html http://localhost/

8080端口配置 00:00:00.000 0.027 2715 GET 200 text/html http://localhost:8080/

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" compressableMimeType="text/html,text/xml,text/plain" compression="on"/>

2.通过我们编程实现.(后面会讲)

后面会讲

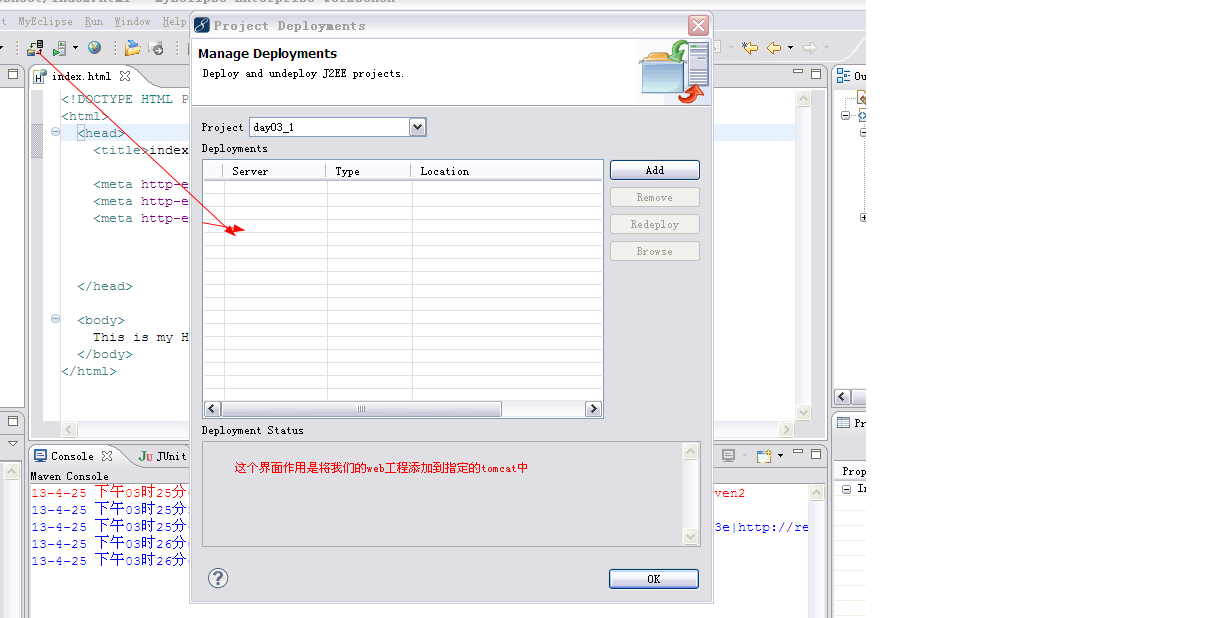
Content-Disposition: attachment; filename=aaa.zip (重要) 文件的下载

//下面三个是禁用浏览缓存

Expires: -1

Cache-Control: no-cache

Pragma: no-cache



4.启动服务器

5.在浏览器中访问web资源.