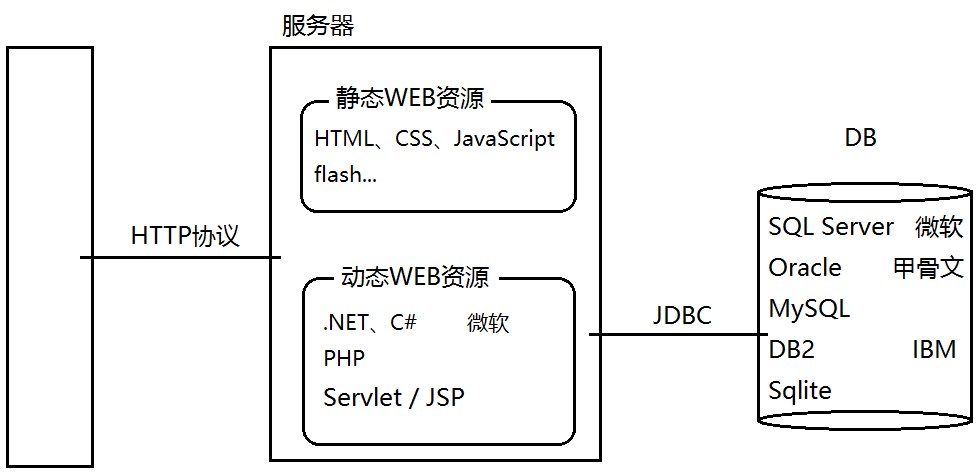
# 8.Tomcat

## 8.1.服务器



动态web资源运行需要服务器环境

客户端发送请求到服务器，服务器调用动态web资源

**Servlet容器、web容器、服务器**

1.1.Servlet容器

java中的动态资源开发技术为Servlet/JSP，但是Servlet、JSP本身不能直接运行，需要运行他们的环境，这样的环境就叫做Servlet容器。

1.2.web容器

能够运行web应用的环境叫做web容器。Servlet容器就是web容器，但是web容器不一定是Servlet容器，因为运行.net、PHP的服务器也是web容器，但是不能运行Servlet

1.3.服务器

服务器从硬件上来说，就是一台电脑，只不过配置较高，而今天，我们所学的是服务器的软件，能够让外界访问，对外界提供服务的就是服务器，服务器就是容器，web容器/Servlet容器都是服务器。

## 8.2.常见的服务器

WebLogic： BEA公司，大型，收费，支持J2EE规范

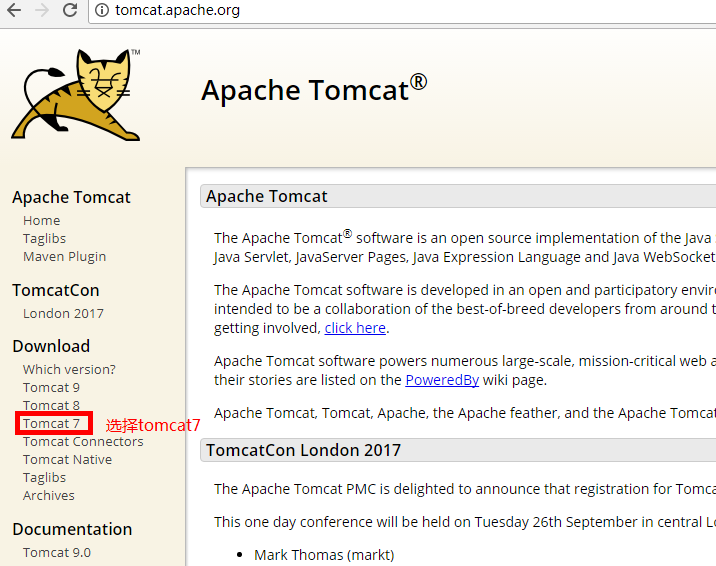
WebSphere：IBM公司，大型，收费，支持J2EE规范

Tomcat ：Apache组织，小型，免费，支持JSP以及Servlet规范

## 8.3.tomcat的下载安装配置

**下载tomcat**

下载地址：<http://tomcat.apache.org/>



下载时请注意，不同版本的区别：

.zip为压缩版，解压后即可使用。

.exe为安装版，安装后使用。



课前资料已经提供

**安装tomcat**

解压

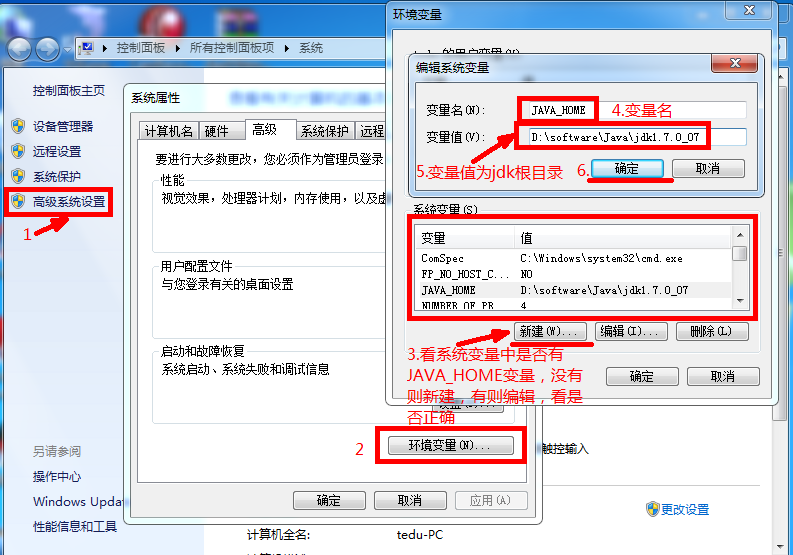
将解压版.zip包拷贝到要安装的目录下，解压，改名为tomcat7.0

注意，安装目录中不允许有空格和中文

配置环境

Tomcat是java写的，需要jdk支持

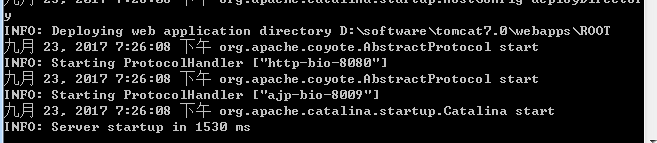
计算机>属性>高级系统设置>环境变量>系统变量>新建>变量名：JAVA\_HOME，变量值：jdk根目录>确定



启动tomcat

通过 [tomcat]/bin/startup.bat文件 启动tomcat服务器

通过 [tomcat]/bin/shutdown.bat文件 关闭tomcat服务器



浏览器访问 http://localhost:8080 如果能够看到tomcat的首页就证明tomcat安装配置成功

安装常见问题

（1）端口占用问题:

1.看是否已经打开tomcat，或是tomcat在关闭时没有正确的关闭，可以运行shutdown.bat命令，然后再次打开

1. 可以在cmd窗口中使用netstat -ano命令查询出哪个程序占用了端口,根据pid，在任务管理器的服务中找到此pid，结束这个程序后再启动tomcat即可

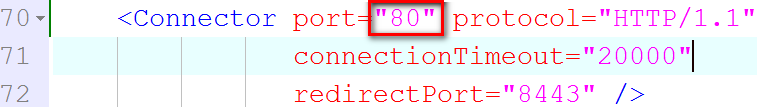
（2）安装路径问题

tomcat安装路径中一定不能有中文和空格，可能一时半会没问题，但是不知道什么时候就可能出现意外。

修改默认的端口号

服务器在启动之后默认监听的是8080端口号, 每次访问时都需要在IP地址或者是主机名后面跟上这个端口号, 如果想省略不写, 可以将8080改为80即可!

找到[tomcat]/conf/server.xml文件并打开，第70行修改如下



重新启动服务器就可以生效！！！

## 8.4.tomcat的目录结构

bin --- 存放tomcat批处理文件的目录

conf --- 存放tomcat配置文件的目录

lib --- 存放tomcat运行时依赖的jar包

logs --- 存放tomcat运行时产生的日志文件的目录

temp --- 存放tomcat运行时产生的临时文件的目录

**webapps --- 这是和开发人员关系最大的目录,是用来存放web应用的目录,我们开发的web资源最终要保存在这个目录下,外界就可以访问了**

work --- 存放tomcat服务器运行时产生的工作文件的目录

## 8.5.web应用

所谓的WEB应用就是将为了实现某一个功能而准备的所有WEB资源, 必须按照一定的目录结构组织起来!

**如何组织一个WEB应用**

news(WEB应用)

|

|-- html, css, js, jsp…(放在WEB应用根目录下的文件可以通过浏览器直接来访问)

|-- WEB-INF 这是一个特殊的目录, 这个目录可以有, 也可以没有, 如果有了这个目录, 这个目录中可以具有如下内容: (注意: 放在这个目录下的资源文件, 不能通过浏览器直接访问!)

|-- classes classes是存放WEB资源中class文件的目录(类似于java工程中的bin目录)

|-- lib lib是存放WEB应用运行时所依赖的jar包的目录

|-- web.xml 该文件是WEB应用的核心配置文件，内容可以在[tomcat]/conf/web.xml中复制

比如，可以建一个文件夹，news，直接在文件夹建几个文件，比如1.html，1.css，1.js等等，也可以将css中的regist文件夹拷贝进来。

**部署web应用到虚拟主机中**

**方式一：**

在[tomcat]/conf/server.xml中的<Host></Host>标签内部, 添加一个Context标签, 配置如下:

|  |
| --- |
| <Context path="/news" docBase="C:\Users\tedu\Desktop\news"/> |

其中 path属性指定WEB应用的虚拟路径，第一个斜杠可以省略

docBase用来指定WEB应用真实路径

可以在浏览器输入相应的虚拟路径网址访问资源，比如

http://localhost/news/regist/regist.html

这种方式每次配置完后都需要重新启动服务器才可以生效, 所以不推荐!!!

**方式二**

在[tomcat]/conf/Catalina/host目录下添加一个xxx.xml文件, 文件的 名字就是WEB应用的虚拟路径, 文件中添加内容如下:

因为文件名中不能有斜杠，可以用#代替斜杠

比如，我们先创建news2文件夹，在[tomcat]/conf/Catalina/localhost目录下添加一个news2.xml文件，文件内容如下

|  |
| --- |
| <Context docBase="C:\Users\tedu\Desktop\news2" /> |

这里的xml可以不添加文档声明，解析器可以解析。

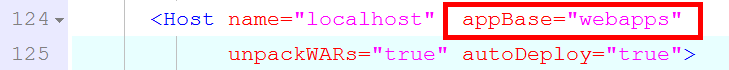
docBase用来指定WEB应用的真实路径。

这种方式配置完后不需要重启服务器就可以起作用!

**方式三**

直接将WEB应用放在localhost主机默认管理的目录下即可！

这里，localhost管理的目录是webapps目录，[tomcat]/conf/server.xml中第124行看到，当然也可以修改，一般都不要动



文件夹的名字成为web应用的虚拟路径。无需重启服务器

比如，创建news3，并放入[tomcat]/webapps中，即可访问

**如何配置缺省的WEB应用?**

第一种配置方式, 只需要将path 的值配置为空(“”)即可。

第二种配置方式, 只需要将xxx.xml文件的名字配置为ROOT.xml即可。

第三种配置方式, 只需要将webapps目录下WEB应用文件夹的名字配置为ROOT即可。

注意: 如果三种方式都配置了缺省WEB应用, 优先级规则是: 第一种 > 第二种 > 第三种

如何配置WEB应用的主页?

在[WEB应用]/WEB-INF目录下, 在web.xml文件中的根标签下添加如下配置：（将1.html设置成主页！！！）

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <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-list>  </web-app> |

## 8.6.虚拟主机

[我的网站](https://www.baidu.com/s?wd=%E6%88%91%E7%9A%84%E7%BD%91%E7%AB%99&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBuHnsrjwWP1-bmWmzrjwB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHRYP1b3" \t "https://zhidao.baidu.com/question/_blank)做好了，自己架设一台[服务器](https://www.baidu.com/s?wd=Web%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBuHnsrjwWP1-bmWmzrjwB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHRYP1b3" \t "https://zhidao.baidu.com/question/_blank)代价也不算小，所以对于个人而言，多个用户共同租用一台服务器——即购买虚拟主机就成了比较现实的选择。

所谓的虚拟主机就是tomcat服务器中的一个站点, tomcat服务器中可以配置多个站点，在访问时就好像访问一台独立的主机一样. tomcat服务器中的一个站点就可以称之为是tomcat服务器模拟出来的一台虚拟主机.

**如何配置虚拟主机**

**第一步：**

在[tomcat]/conf/server.xml文件的Engine标签内部添加一个Host标签, 配置如下:（同时在[tomcat]/目录下创建baidu文件夹，并放入一个web应用）

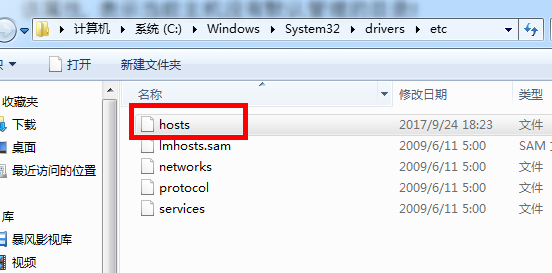
|  |
| --- |
| <Host name="www.baidu.com" appBase="baidu"></Host> |

其中name是必须存在的属性, 用来指定虚拟主机的名称.

appBase是可选属性, 用来指定虚拟主机默认管理的目录, 如果没有配置该属性, 表示当前主机没有默认管理的目录!（则无法用第三种方式部署web应用）

**第二步：**

配置完后, 还需要在DNS服务器中配置主机名和IP地址的映射关系, 但是DNS服务器一般不能修改, 可以通过hosts文件进行模拟, 在下面的路径C:\Windows\System32\drivers\etc中找到hosts文件。



在最后一行加入如下配置：（IP和网址之间是Tab键）

|  |
| --- |
| 127.0.0.1 www.baidu.com |

在浏览器输入路径，如：www.baidu.com/news3/1.html

注意：当可以上网的时候，很有可能你输入的网址不能生效，直接去访问百度去了，这时候需要清dns缓存，打开cmd窗口，输入ipconfig/flushdns，浏览器清缓存再次打开即可

**如何配置默认虚拟主机**

当通过IP访问服务器时，会访问哪台主机呢？

在[tomcat]/conf/server.xml文件的Engine标签上有一个属性defaultHost，可以通过此属性改变默认虚拟主机，比如（默认是localhost）

|  |
| --- |
| <Engine name="Catalina" defaultHost="www.baidu.com"> |

**综合练习**

配置为www.163.com虚拟主机, 以三种方式配置web应用,并选择一种配置为缺省web应用, 并且配置web应用的主页, 最终实现直接访问www.163.com能够显示主页的内容！！！

## 8.7.其他相关

**打war包**

为什么要打war包？

第一：体积小，方便传输。

第二：将war包直接放进虚拟主机默认管理目录，会自动解压并发布。

**方式一：**

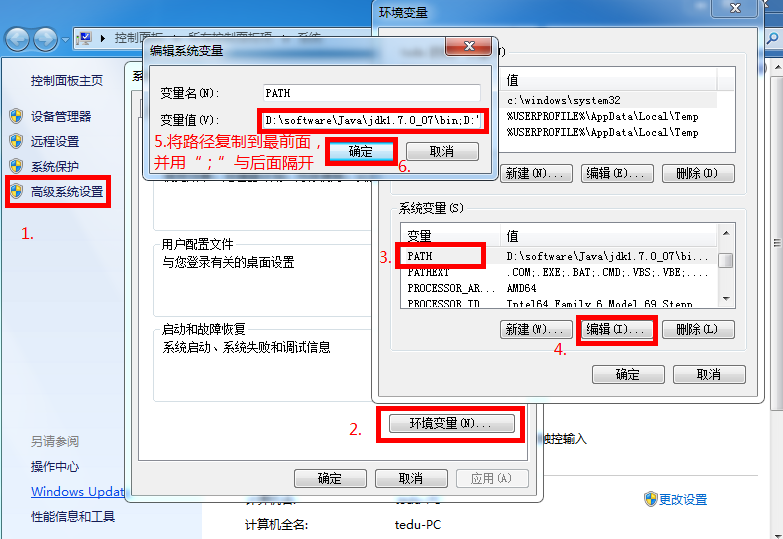
在cmd窗口, 先进入到WEB应用的根目录, 通过 “jar -cvf xx.war \*” 命令, 可以将当前目录下所有的文件及目录打成一个war包.



然后将war包放入localhost管理的目录下，会自动解压发布WEB应用。放入baidu下不会生效，因为我们只配置了最基本的信息。

**如果此命令不生效，可以手动配置path路径**

先找到jdk目录下的bin目录，如D:\software\Java\jdk1.7.0\_07\bin复制此目录，然后点击计算机>属性，操作如下图



操作完成后，重新打开cmd窗口，敲jar命令。

**方式二：**

直接将文件内容压缩为zip包，将包名改为xxx.war即可，然后将war包放入localhost管理的目录下（注意不能有中文）

**通用的web.xml文件**

在[tomcat]/conf/目录下的web.xml文件, 是一个通用的web.xml文件, 这个web.xml文件的内容, 将会被所有WEB应用web.xml文件所”继承”。

比如刚安装tomcat的时候，访问localhost直接会进入tomcat主页，而tomcat中WEB-INF下的xml中并未配置，因为继承了通用web.xml

## 8.8.HTTP协议

**什么是HTTP协议?**

HTTP是hypertext transfer protocol（超文本传输协议）的简写，用来规定浏览器客户端和服务器之间进行通信的方式

**三个基本原则**

（1）基于请求响应模型

（2）一次请求对应一次响应

（3）请求只能由客户端发出, 服务器只能被动的等待请求, 做出响应.

**HTTP协议详解**

**http请求**

HTTP请求信息由3部分组成：

1.请求行：包含 请求方式 请求的资源 所遵循的协议

2.请求头： http中的请求头非常多

3.实体内容： 需要传递给服务器的内容

**请求行：GET /news3/1.html HTTP/1.1**

**请求方式：**一共有7种请求方式,但真正使用的只有两种GET、POST。

POST和GET之间的区别:

GET：请求请求参数位于URL后面,所能传输的请求参数最大不超过1KB,安全性比较低。

POST：请求参数位于请求的实体内容中,没有大小限制,也比较安全

只有当一个表单method明确被指定为POST时,才是POST提交,其他情况下都是GET提交.

**请求的资源：**/news3/1.html

**所遵循的协议：**HTTP/1.1

**请求头：**

http协议中请求头非常多，下面列出常见的请求头及其功能：

!Host: www.tedu.cn:80

-- 需要访问的虚拟主机的名称

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) -- 客户端的基本信息

Accept: text/html,image/\*

-- 通知服务器当前浏览器可以接受那些格式的数据

Accept-Language: en-us,zh-cn

-- 浏览器接受的语言环境,和国际化相关的头

Accept-Encoding: gzip,compress

-- 浏览器可以接受的压缩格式

Accept-Charset: ISO-8859-1

-- 浏览器可以接受的字符集编码

**!!Referer: http://www.tedu.cn/index.jsp**

-- 指是从哪个页面跳转过来的，这是和防盗链相关的头。

**!!!Cookie**

-- 和cookie相关的头

Connection: close/Keep-Alive

-- 是否继续保持连接

Date: Fri, 17 Sep 2017 18:23:51 GMT

-- 当前发送请求的时间

**实体内容：**

当post提交时，会存在参数，就是实体内容

比如将注册页面regist.html中form表单上添加如下属性

|  |
| --- |
| <form method="post" action="http://localhost/news3/1.html"> |

然后注册用户，在可以找到参数

**http响应**

HTTP响应信息由3部分组成：

1.状态行：包含 所遵循的协议 状态码

2.响应头： http中的响应头非常多

3.实体内容： 服务器发送回的内容

**状态行：HTTP/1.1 200 OK**

**所遵循的协议：HTTP/1.1**

**状态码：**

状态码, 就是一个三位的数值, 用来表示服务器对请求处理的结果

常见的状态码:

200: 表示服务器成功的处理该请求

302: 和location配合实现请求重定向(后面讲!)

304/307: 通知浏览器使用缓存

404: 表示请求的资源不存在, 客户端错误

500: 服务器处理请求时出现了错误!

**响应头：**

http协议中响应头非常多，下面列出常见的响应头及其功能：

**!!!Location: http://www.tedu.cn/index.jsp**

-- 配合302使用实现请求重定向

Server:apache tomcat

-- 服务器的基本信息

Content-Encoding: gzip

-- 服务器发送的数据使用的压缩格式

Content-Length: 80

-- 服务器发送的数据的大小

Content-Language: zh-cn

--服务器发送的数据使用的语言环境 国际化相关的头

**!Content-Type: text/html; charset=GB2312**

-- 服务器发送的数据是什么格式的,如果是字符格式的数据,则还可以通知服务器发送的数据使用的是什么编码,浏览器会根据这个头指定的编码决定使用哪个编码来打开收到的数据

**!Refresh: 1;url=http://www.tedu.cn**

-- 定时刷新相关的头,通知浏览器,过几秒后自动刷新访问哪个地址

Content-Disposition: attachment;filename=aaa.zip

-- 通知浏览器以附件的形式打开发送过去的数据,是和文件下载相关的头

**!!!Set-Cookie:SS=Q0=5Lb\_nQ; path=/search**

-- 和Cookie相关的头

**!Expires: -1**

-- 通知浏览器是否缓存

**!Cache-Control: no-cache**

-- 通知浏览器是否缓存

**!Pragma: no-cache**

-- 通知浏览器是否缓存

--之所以一个功能有三个头,是因为历史原因.所以为了网页的兼容性考虑,通常这三个头要一起使用

Connection: close/Keep-Alive

-- 是否保持连接

Date: Fri, 17 Feb 2017 18:24:11 GMT

-- 响应时的时间

**实体内容：**

就是服务器响应的内容。

## 8.9.作业练习

1. Tomcat默认端口是多少，在哪里修改？

答：默认8080，可以在tomcat/conf/server.xml Connector 中的 prot 将tomcat监听的端口号从8080 改为 80

1. 应用服务器有哪些

答：WebLogic： BEA公司，大型，收费，支持全部J2EE规范

WebSphere：IBM公司，大型，收费，支持全部J2EE规范

Tomcat ：Apache组织，小型，免费，支持JSP以及Servlet规范

1. 什么是http协议？

答：超文本传输协议，用来规定浏览器客户端和服务器之间进行通信的方式。

1. 状态码及表示的含义？

答：200: 表示服务器成功的处理该请求

302: 和location配合实现请求重定向

304/307: 通知浏览器使用缓存

404: 表示请求的资源不存在, 客户端错误

500: 服务器处理请求时出现了错误!

1. HTTP协议工作原则？

答：基于请求响应模型

一次请求对应一次响应

请求只能由客户端发出, 服务器只能被动的等待请求, 做出响应.