

一、WEB开发入门

2018年9月20日 9:19

1. WEB开发的相关知识

- 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、CSS、JavaScript
- 常用动态web资源开发技术：
 - JSP/Servlet、ASP、PHP等
 - 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。

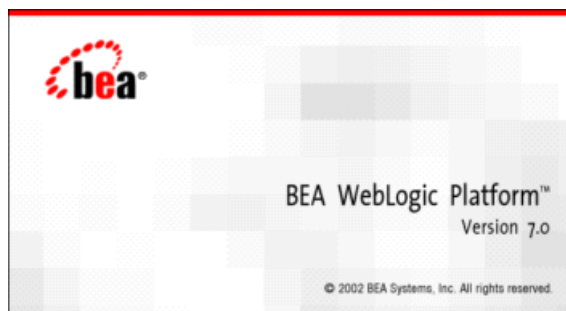
2. WEB服务器

学习web开发，需要先安装一台web服务器，然后再在web服务器中开发相应的web资源，供用户使用浏览器访问。

注意：这里所说的服务器不是指服务器硬件资源，而是指服务器软件。

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



二、Tomcat

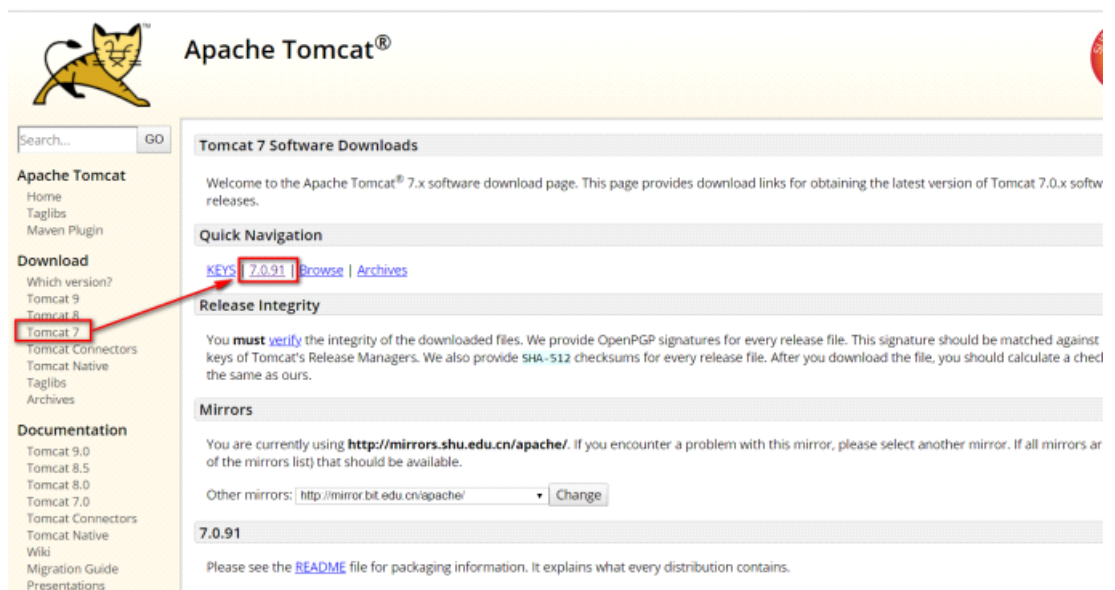
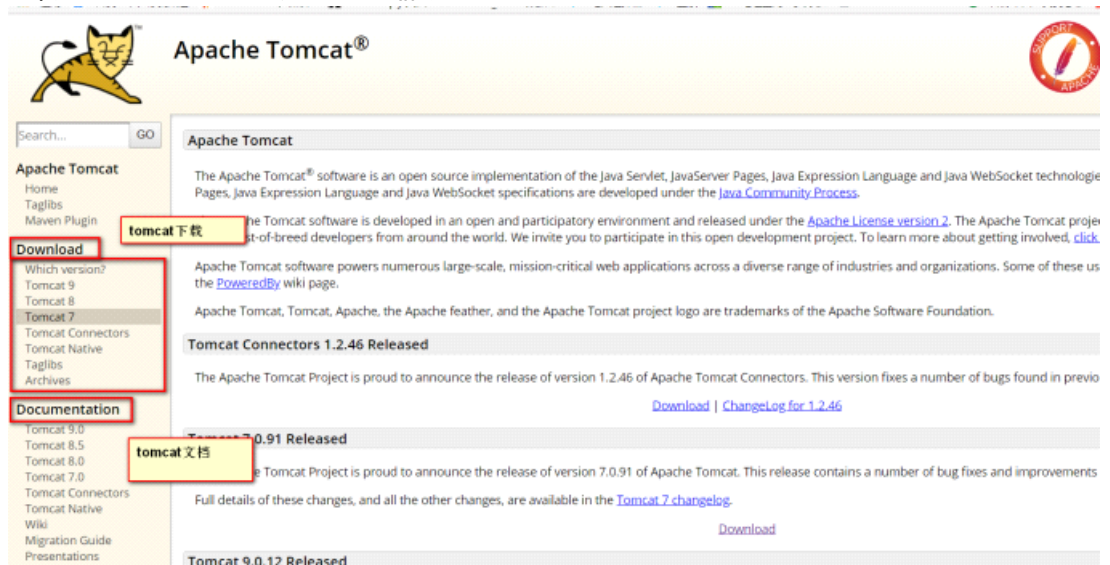
2018年9月20日 14:53

1. TOMCAT服务器的安装与配置

常见服务器：WebLogic（BEA）、webSphere（IBM）、Tomcat（Apache）

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





■ 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

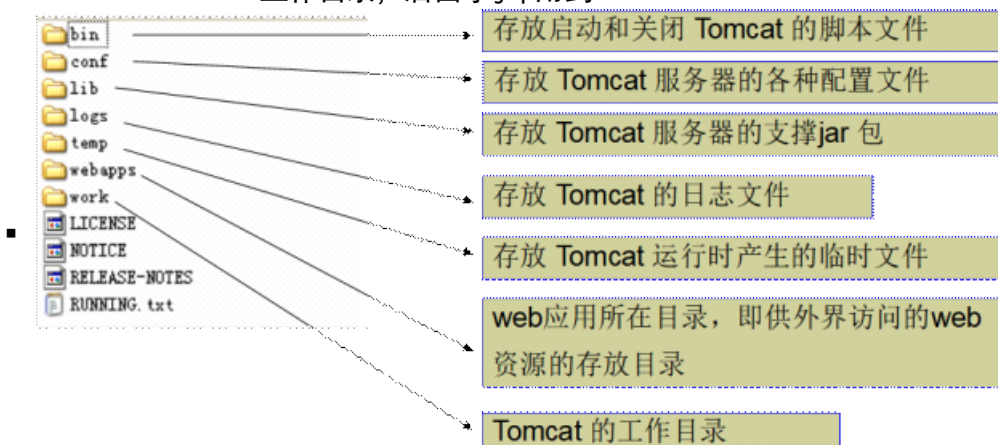
在浏览器端输入 <http://localhost:8080/>

显示如下界面代表安装成功



b. Tomcat的目录结构

- bin--存放tomcat启动关闭所用的批处理文件
- conf--tomcat的配置文件，最终要的是server.xml
 - *实验:修改servlet.xml,更改tomcat运行所在的端口号,从8080改为80
- lib--tomcat运行所需jar包
- logs--tomcat运行时产生的日志文件
- temp--tomcat运行时使用的临时目录,不需要我们关注
- webapps--web应用所应存放的目录
- work--tomcat工作目录,后面学jsp用到



三、虚拟主机及web应用

2018年9月28日 16:50

虚拟主机概念:

一个tomcat可以认为是一台真实主机. 在一台真实主机中可以配置多个站点, 这些站点在访问者看来访问他们就像在访问各自独立的主机一样, 所以我们可以认为这些站点都运行在tomcat这台真实主机当中的各自的虚拟主机当中. 一个网站就可以认为是一个虚拟主机。

web应用概念:

一个虚拟主机中会有很多的web资源, 但是web资源不能直接交给虚拟主机管理, 需要按照一定方式组织成web应用虚拟主机才能使用. 一般来说我们会按照功能将某一功能相关的所有的web资源组织成一个web应用后再交给虚拟主机。

1. web应用

- WEB应用程序指供浏览器访问的程序, 通常也简称为web应用, 是为了提供某一特定功能而按照一定方式组织起来的web资源的组合。
- 一个web应用由多个静态web资源和动态web资源组成, 如:
 - html、css、js文件
 - Jsp文件、java程序、支持jar包、
- 配置文件
 - 一个web应用所使用的web资源我们通常使用目录进行组织, 这个目录我们通常称为 web应用所在的目录。Web应用开发好后, 若想供外界访问, 需要把web应用所在目录交给web服务器管理, 这个过程称之为虚拟目录的映射。

2. 配置web应用及虚拟目录映射

- (1) 在server.xml的<Host>标签下配置<Context path="虚拟路径" docBase="真实路径" />如果path=""则这个web应用就被配置为了这个虚拟主机的默认web应用。
- (2) 在tomcat/conf/引擎名/虚拟主机名 之下建立一个.xml文件, 其中文件名用来指定虚拟路径, 如果是多级的用#代替/表示, 文件中配置<Context docBase="真实目录" />, 如果文件名起为ROOT.xml则此web应用为默认web应用。
- (3) 直接将web应用放置到虚拟主机对应的目录下, 如果目录名起为ROOT则此web应用为默认web应用
~如果三处都配置默认web应用则server.xml > config/.../xx.xml > webapps。

3. web应用的目录结构

开发web应用时, 不同类型的文件有严格的存放规则, 否则不仅可能会使web应用无法访问, 还会导致web服务器启动报错。

- web应用目录结构: 如果目录结构不合法可能会出现各种奇葩的问题。

```
news
|
|--静态资源和JSP文件都可以直接放置在web应用的目录下, 直接放在web应用下的内容, 浏览器可以直接访问到
|
|--WEB-INF -- 可以没有, 但是最好有, 如果有则一定要保证他的目录结构是完整的. 放置在WEB-INF目录下的所有资源浏览器没有办法直接进行访问
|   |--classes -- 动态web资源运行时的class文件要放在这个目录下
|   |--lib -- 动态web资源运行时所依赖的jar包要放在这个目录下
|   |--web.xml -- 整个web应用的配置文件, 配置主页/Servlet的映射/过滤器监听器的配置都需要依赖这个文件进行
```

- web应用中, web.xml文件是其中最重要的一个文件, 它用于对web应用中的web资源进行配置。

4. web.xml文件的作用:

- 通过web.xml文件, 可以将web应用中的:
 - 某个web资源配置为web应用首页
 - 将servlet程序映射到某个url地址上
 - 为web应用配置监听器
 - 为web应用配置过滤器
 - 但凡涉及到对web资源进行配置, 都需要通过web.xml文件
- 举例: 通过web.xml文件配置web应用首页再配置成网站主页。
- 注意: Web.xml文件必须放在web应用\WEB-INF目录下。
*实验: 配置一个web应用的主页

5. 配置虚拟主机

- 在server.xml中<Engine>标签下配置<Host>, 其中name属性指定虚拟主机名, appBase指定虚拟主机所在的目

录

- 只在servlet.xml中配置Host标签，还不能使其他人通过虚拟主机名访问网站，还需要在DNS服务器上注册一把，我们可以使用hosts文件模拟这个过程
 - Host文件路径：C:\Windows\System32\drivers\etc\hosts
 - 使用hosts文件原因：由于我们在浏览器访问的地址，会通过DNS服务器将复制翻译成对应的ip地址才能找到服务器，这其中的翻译过程是由DNS服务器完成的，我们做实验的时候没有办法修改DNS服务器，所以修改本地的hosts文件来实现类似的操作。
- 默认虚拟主机：在配置多个虚拟主机的情况下，如果浏览器使用ip地址直接访问网站时，该使用哪个虚拟主机响应呢？可以在<Engine>标签上设置defaultHost来指定。
- <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
errorReportValveClass	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	否

6. 杂项

- 打war包：方式一：jar -cvf news.war * 方式二：直接用压缩工具压缩为zip包，该后缀为.war
- 通用context和通用web.xml，所有的<Context>都继承子conf/context.xml，所有的web.xml都继承自conf/web.xml

7. 作业

配置一个www.google.com这样一台虚拟主机，其中包含news应用和mail应用，news应用中有一个new.html。实现在浏览器中输入www.google.com直接看到new.html中的网页信息。

四、HTTP协议

2018年9月21日 14:47

1. 什么是HTTP协议

- 客户端连上web服务器后，若想获得web服务器中的某个web资源，需遵守一定的通讯格式，HTTP协议用于定义客户端与web服务器通讯的格式。
- 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。
- ~OSI网络七层协议
应用层（HTTP、FTP、SMTP、POP3、TELNET）→表示层→会话层→传输层（TCP、UDP）→网络层（IP）→数据链路层→物理层

2. HTTP协议简介

- HTTP使用请求-响应的方式进行传输，一个请求对应一个响应，并且请求只能是由客户端发起的。
- 利用Telnet演示请求与响应的过程
 - **Telnet服务打开方式详见资料目录。**
- HTTP协议的版本：HTTP/1.0、HTTP/1.1

3. HTTP1.0和HTTP1.1的区别

- 在HTTP1.0协议中，客户端与web服务器建立连接后，在请求得到一次响应后，获取web资源后连接断开。只能获得一个web资源。
- HTTP1.1协议，允许客户端与web服务器建立连接后，在请求得到一次响应后，仍然等待下一次请求，直到一段时间后，没有请求发送，连接断开。在一个连接上获取多个web资源。
- 利用telnet演示HTTP1.0和HTTP1.1的区别
 - 首先准备一个web应用
 - 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。
- 一个好多同学搞不清楚的问题：
 - 一个web页面中，使用img标签引用了三幅图片，当客户端访问服务器中的这个web页面时，客户端总共会访问几次服务器，即向服务器发送了几次HTTP请求。

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、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/links.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 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 和cookie相关的头
- 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)、以及实体内容。

● 举例：

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 13 Jul 2000 05:46:33 GMT
Content-Length: 2291
Content-Type: text/html
Cache-control: private
```

```
<HTML>
<BODY>
.....
```

← 状态行

← 多个响应头 —

← 一个空行

← 实体内容 —

状态行用于描述服务器对请求的处理结果。

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。

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

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实现请求转发
- 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

拓展：JavaEE规范

2018年9月28日 14:24

java的大方向就是JavaEE，JavaEE不仅仅是socket编程，具体包括13中核心技术。

JAVAE的核心API与组件：

JAVAE平台由一整套服务（Services）、应用程序接口（APIs）和协议构成，它对开发基于Web的多层应用提供了功能支持，下面对JAVAE中的13种技术规范进行简单的描述(限于篇幅，这里只进行简单的描述)：

1、JDBC(Java Database Connectivity) JDBC API为访问不同的数据库提供了一种统一的途径，象ODBC一样，JDBC对开发者屏蔽了一些细节问题，另外，JDBC对数据库的访问也具有平台无关性。

2、JNDI(Java Name and Directory Interface) JNDI API被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源如DNS和LDAP，本地文件系统，或应用服务器中的对象。

3、EJB(Enterprise JavaBean) JAVAE技术之所以赢得媒体广泛重视的原因之一就是EJB。它们提供了一个框架来开发和实施分布式商务逻辑，由此很显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。EJB规范定义了EJB组件在何时如何与它们的容器进行交互作用。容器负责提供公用的服务，例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是，EJB并不是实现JAVAE的唯一途径。正是由于JAVAE的开放性，使得有的厂商能够以一种和EJB平行的方式来达到同样的目的。

4、RMI(Remote Method Invoke) 正如其名字所表示的那样，RMI协议调用远程对象上方法。它使用了序列化方式在客户端和服务端传递数据。RMI是一种被EJB使用的更底层的协议。

5、Java IDL/CORBA 在Java IDL的支持下，开发人员可以将Java和CORBA集成在一起。他们可以创建Java对象并使之可在CORBA ORB中展开，或者他们还可以创建Java类并作为和其它ORB一起展开的CORBA对象的客户。后一种方法提供了另外一种途径，通过它Java可以被用于将你的新的应用和旧的系统相集成。

6、JSP(Java Server Pages) JSP页面由HTML代码和嵌入其中的Java代码所组成。服务器在页面被客户端所请求以后对这些Java代码进行处理，然后将生成的HTML页面返回给客户端的浏览器。

7、Java Servlet Servlet是一种小型的Java程序，它扩展了Web服务器的功能。作为一种服务器端的应用，当被请求时开始执行，这和CGI Perl脚本很相似。Servlet提供的功能大多与JSP类似，不过实现的方式不同。JSP通常是大多数HTML代码中嵌入少量的Java代码，而servlets全部由Java写成并且生成HTML。

8、XML(Extensible Markup Language) XML是一种可以用来定义其它标记语言的语言。它被用来在不同的商务过程中共享数据。XML的发展和Java是相互独立的，但是，它和Java具有的相同目标正是平台独立性。通过将Java和XML的组合，您可以得到一个完美的具有平台独立性的解决方案。

9、JMS(Java Message Service) JMS是用于和面向消息的中间件相互通信的应用程序接口(API)。它既支持点对点的域，有支持发布/订阅(publish/subscribe)类型的域，并且提供对下列类型的支持：经认可的消息传递,事务型消息的传递，一致性消息和具有持久性的订阅者支持。JMS还提供了另一种方式来对您的应用与旧的后台系统相集成。

10、JTA(Java Transaction Architecture) JTA定义了一种标准的API，应用系统由此可以访问各种事务监控。

11、JTS(Java Transaction Service) JTS是CORBA OTS事务监控的基本的实现。JTS规定了事务管理器的实现方式。该事务管理器是在高层支持Java Transaction API (JTA)规范，并且在较底层实现OMG OTS specification的Java映像。JTS事务管理器为应用服务器、资源管理器、独立的应用以及通信资源管理器提供了事务服务。

12、JavaMail JavaMail是用于存取邮件服务器的API，它提供了一套邮件服务器的抽象类。不仅支持SMTP服务器，也支持IMAP服务器。

13、JAF(JavaBeans Activation Framework) JavaMail利用JAF来处理MIME编码的邮件附件。MIME的字节流可以被转换成Java对象，或者转换自Java对象。大多数应用都可以不需要直接使用JAF

- 问题：在配置web应用映射的时候，配置多个缺省会导致在访问localhost的时候出现404
原因：配置多个缺省，缺省先后顺序为，server.xml > catalina/localhost/ROOT.xml > ROOT目录
- 问题：更新页面依然出现乱码
原因：缓存问题，清楚浏览器缓存。
- 问题：双击startup.bat闪退
原因：1.查看java环境变量，JAVA_HOME对应的路径不能有";"，而且要是一个真实存在的目录，JAVA_HOME环境变量所对应的目录，指向java目录而不是java/bin目录
2.如果在配置过server.xml过后双击startup.bat闪退，基本确定在配置server.xml的时候出现错误，检查有无"<>"，"/"等符号缺失，也可以打开tomcat/logs/catalina.log进行查看日志信息，判断是什么错误。
- 问题：PID为4的系统进程占用80端口
一、先使用netstat命令查看一下80端口是被谁占用了。

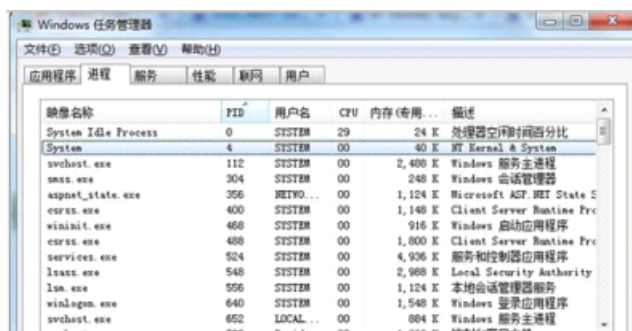
1、使用电脑快捷组合键Win+R打开快速运行，在运行窗口中输入“cmd”命令点击回车确定调出管理员命令提示符，紧接着输入“netstat -ano | more”。如下图所示：



2、点击回车后，显示出一大段文字，找到“本地地址”为0.0.0.0:80的一行，在行的最后我们会看到PID为“4”。记住此PID；如下图所示：



3、打开Windows任务管理器，在进程中找到PID为“4”的程序，如下图所示：显示80端口被pid4 system占用。



二、分析端口被占用原因:

该进程是Http.sys。它是http API的驱动组件，Http栈服务器。如果该端口被Http.sys占用，说明一些正在使用http.sys的应用程序在运行。通过netsh http show servicestate 命令能找出是哪个应用程序在使用Http.sys。

三、解决问题:可以禁用http.sys

禁用http.sys进程可以通过命令提示符来完成，首先输入net stop http，然后确认继续操作，在输入Sc config http start= disabled即可停止该项服务。如下图：



```
管理员: C:\Windows\system32\cmd.exe
C:\Users\David>net stop http
下面的服务依赖于 HTTP 服务。
停止 HTTP 服务也会停止这些服务。

Web Management Service
Windows Remote Management (WS-Management)
Windows Event Collector
World Wide Web Publishing Service
BranchCache
IIS Admin Service
HomeGroup Provider
Function Discovery Resource Publication
PnP-X IP Bus Enumerator
Function Discovery Provider Host

你想继续此操作吗? (Y/N) [N]: y

管理员: C:\Windows\system32\cmd.exe
C:\Users\David>Sc config http start= disabled
[SC] ChangeServiceConfig 成功
```