



# JavaWeb开发入门



# 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
- 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。

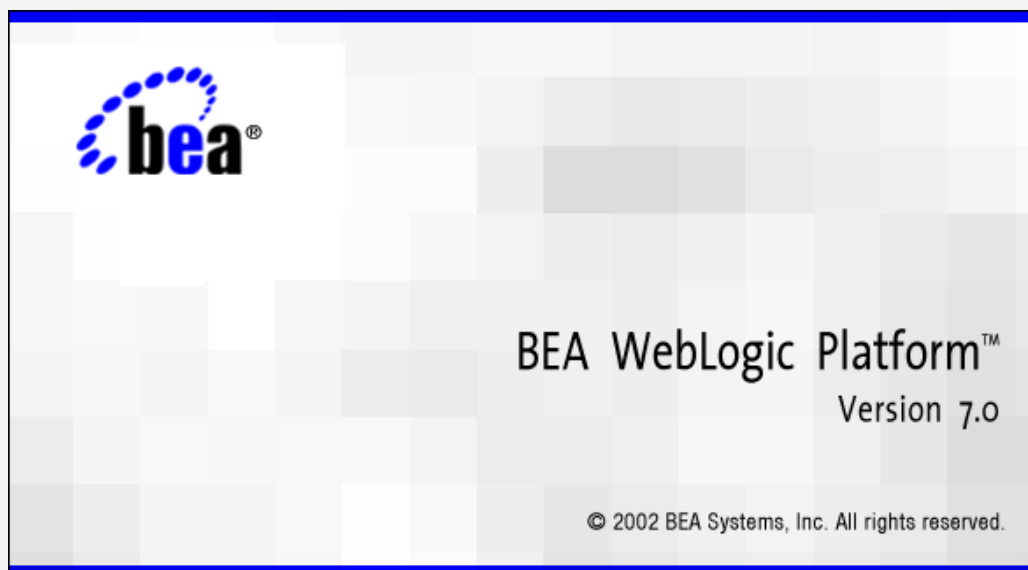
## WEB服务器

- 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。

```
ServerSocket server = new ServerSocket(9999);  
Socket socket = server.accept();  
OutputStream out = socket.getOutputStream();  
InputStream in = new FileInputStream("D:a.html");  
int i=0;  
byte b[] = new byte[1024];  
while((i=in.read(b))!=-1){  
    out.write(b, 0, i);  
}  
in.close();  
out.close();
```

## 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JavaEE规范，而且不断的完善以适应新的开发要求，启动界面如图



## 常见WEB服务器

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



## 常见WEB服务器

- 大名鼎鼎的红帽的产品，原来属于开源组织JbossAS，后被红帽收购。支持JavaEE规范，免费。



## 常见WEB服务器

- 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JSP以及Servlet规范。启动界面如图



# Tomcat简介

- Tomcat是在SUN公司推出的小型Servlet/JSP调试工具的基础上发展起来的一个优秀的Servlet容器，Tomcat本身完全用Java语言编写。目前是Apache开源软件组织的一个软件项目，它的官方网址为<http://tomcat.apache.org>。得到了广大开源代码志愿者的大力支持，且可以和目前大部分的主流Web服务器(IIS、Apache服务器)一起工作，它运行稳定、可靠且高效。已成为目前开发企业JavaWeb应用的最佳Servlet容器选择之一。
- 我们后面的JavaWeb学习都是基于Tomcat服务器的。



## Tomcat服务器

- **Tomcat官方站点:**  
<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x86  
x64
  - ✓ zip文件是Windows系统下的压缩版本 (建议)

# Tomcat的版本

- Tomcat的版本随着SUN公司推出的Servlet/JSP规范不同而不同，基本上是SUN公司每推出一个版本，Tomcat也会发行新的版本以适应新的规范。

Tomcat版本	Servlet/JSP规范	JDK版本
7.0	3.0/2.2	1.6
6.0.x	2.5/2.1	1.5
5.5.x	2.4/2.0	1.4
5.0.x	2.4/2.0	1.4
4.1.x	2.3/1.2	1.3
3.3.x	2.2/1.1	1.1

## 安装Tomcat

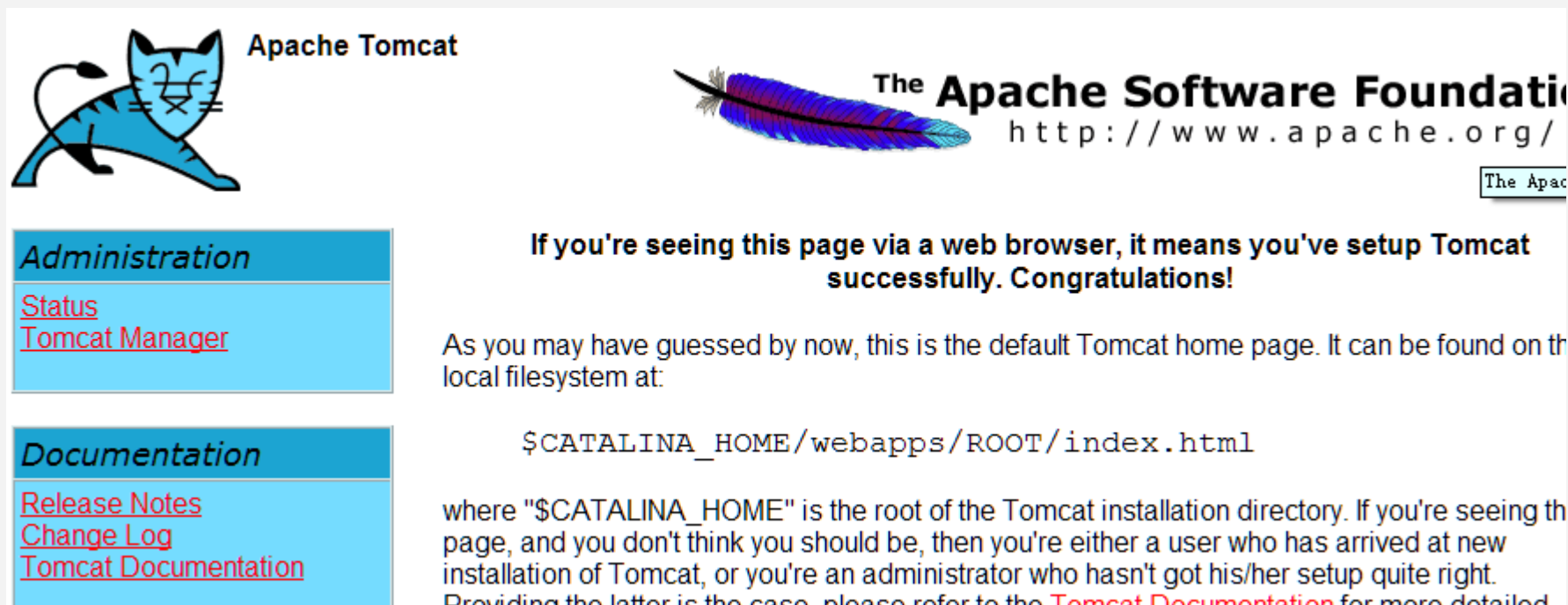
- Tomcat6本身是基于JDK1.5的应用程序，因此在安装和使用之前必须先行安装1.5及以上版本的JDK(JRE)
- 解压安装
- Exe安装

# Tomcat启动方式

- 使用Tomcat服务程序(安装版)
- 使用Tomcat程序组中的快捷菜单(安装版)
- 使用Tomcat.exe程序(安装版)
- 使用批处理程序(使用解压版)

## 测试Tomcat服务器

- 双击 bin 目录下的 startup.bat 文件
- 输入 `http://localhost:8080/`，显示如下界面代表安装成功



# 常见启动问题



## Java\_home环境变量

- 在windows中设置java\_home环境变量
- 只要在setclasspath.bat批处理文件第一次使用JAVA\_HOME环境变量之前的任何地方，将JAVA\_HOME环境变量设置为JDK的主目录，就可以使用startup.bat文件启动Tomcat了。

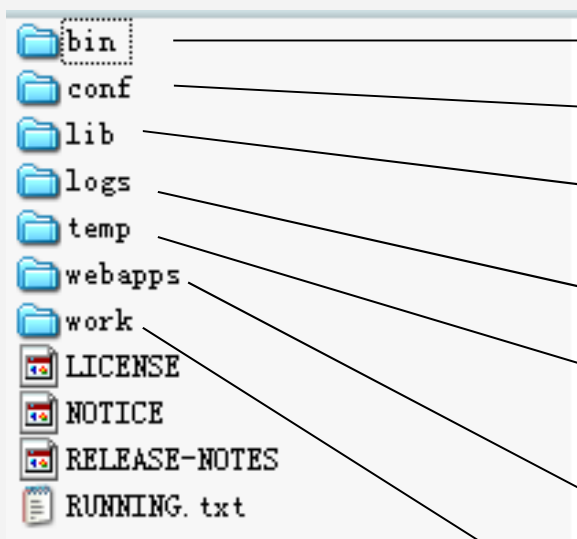
## 端口占用问题(更改默认端口)

## Catalina\_home环境变量的设置问题



## Tip: Tomcat 目录结构

- Tomcat 的目录层次结构



存放启动和关闭 Tomcat 的脚本文件

存放 Tomcat 服务器的各种配置文件

存放 Tomcat 服务器的支撑jar 包

存放 Tomcat 的日志文件

存放 Tomcat 运行时产生的临时文件

web应用所在目录，即供外界访问的web  
资源的存放目录

Tomcat 的工作目录

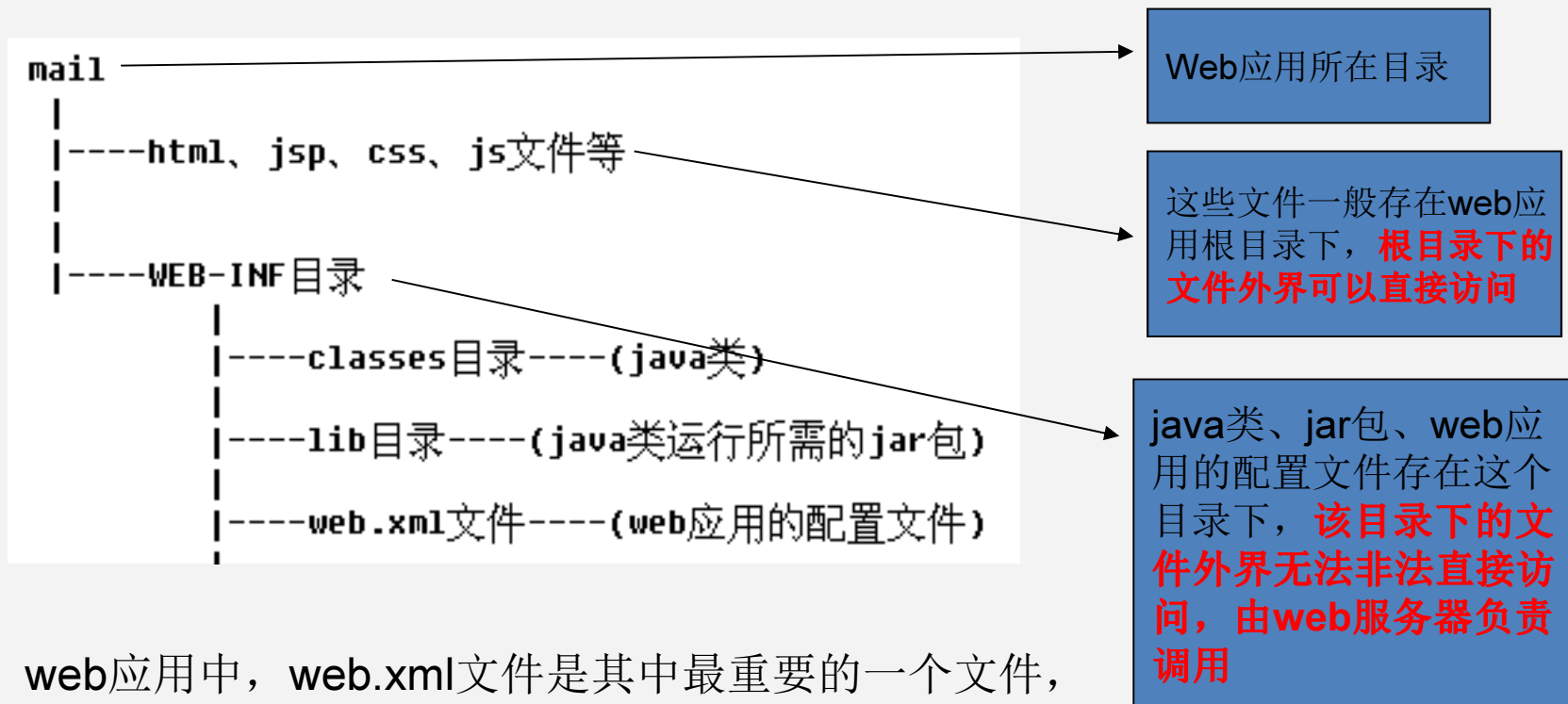
# JavaWEB应用程序

- WEB应用程序指供浏览器访问的程序，通常也简称为web应用。
- 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
  - .....
- 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。



## JavaWEB应用的组成结构

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。

# 发布JavaWeb应用

- 演示建立一个web应用
- 发布JavaWeb应用(开放式目录和war)

# Web组件的URL

- 无论是开放式目录结构还是打包文件方式发布web应用，web应用的默认URL入口都是Web应用的根目录名。例如要访问MyApp应用，它的URL入口为/MyApp，如访问本地服务

<http://localhost:8080/MyApp>

(http://127.0.0.1:8080/MyApp)

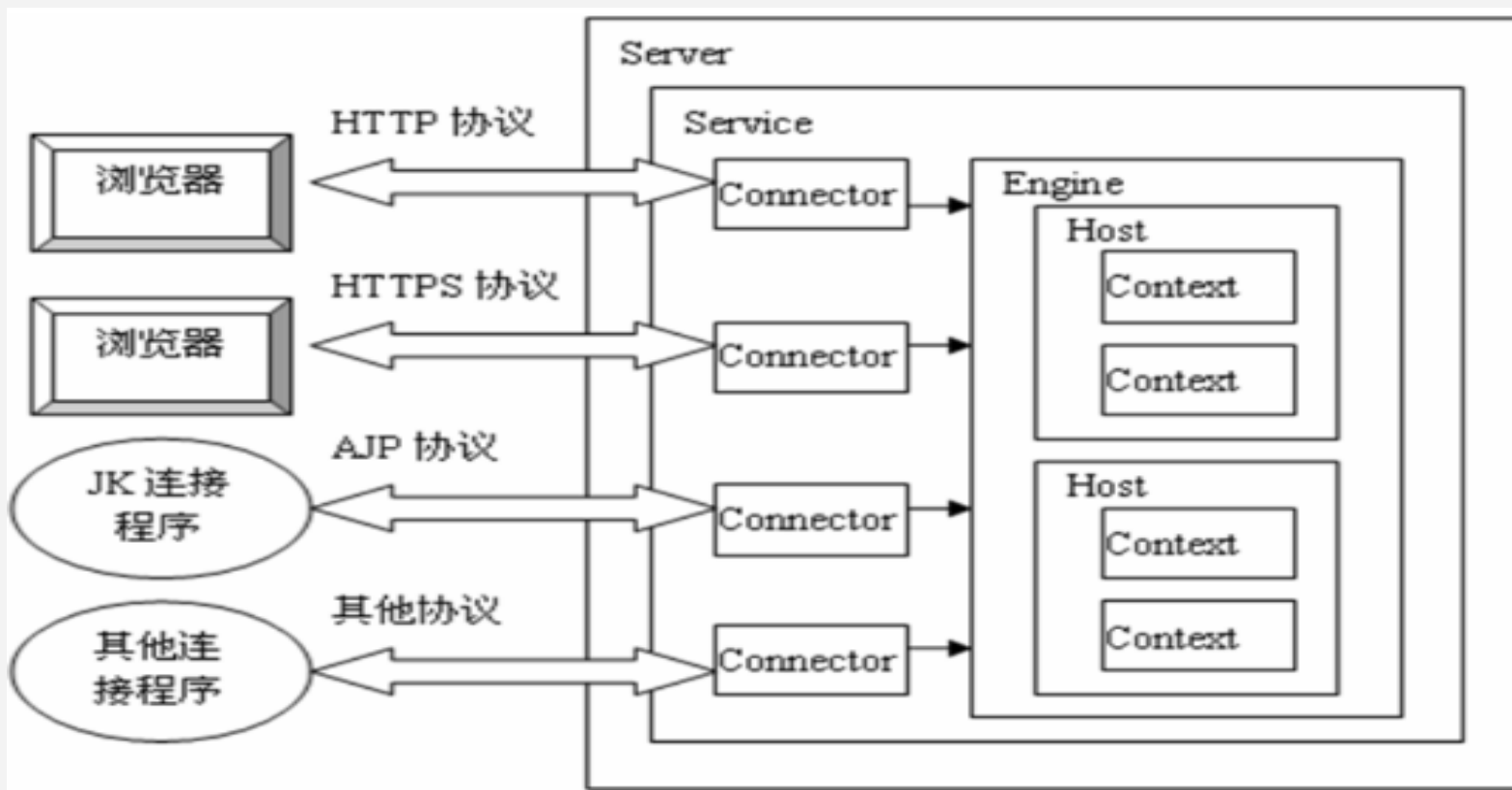
# web.xml文件

- 通过web.xml文件，可以将web应用中的：
  - 某个web资源配置为网站首页
  - 将servlet程序映射到某个url地址上
  - .....
  - 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文件配置网站首页。
- 注意：web.xml文件必须放在web应用\WEB-INF目录下。

# Tomcat的组成结构

- Tomcat本身由一系列可配置的组件构成，其中核心组件是Servlet容器组件，它是所有其他Tomcat组件的顶层容器。每个组件都可以在Tomcat安装目录  
/conf/server.xml文件中进行配置，每个Tomcat组件在server.xml文件中对应一种配置元素。以下用XML的形式展示了各种Tomcat组件之间的关系

## Tomcat体系架构



# Tomcat的组成结构

**<Server>** 代表整个 Servlet 容器组件，是最顶层元素。可以包含一个或多个<Service>元素↵

**<Service>** 包含一个<Engine>元素以及一个或多个<Connector>元素，这些<Connector>共享同一个<Engine>↵

**<Connector/>** 代表和客户程序实际交互的组件，负责接收客户请求，以及向客户返回响应↵

**<Engine>** 每个<Service>元素只能包含一个<Engine>元素。它处理在同一个<Service>中所有<Connector>接收到的客户请求↵

**<Host>** 在一个<Engine>中可以包含多个<Host>，它代表一个虚拟主机(网站)，它可以包含一个或多个 Web 应用↵

**<Context/>** 使用最频繁的元素。代表了运行在虚拟主机上的单个 Web 应用。↵

**</Host>**↵

**</Engine>**↵

**</Service>**↵

**</Server>**↵

## 虚拟目录的映射方式

- 在server.xml文件的host元素中配置，例如：

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true"
      xmlValidation="false" xmlNamespaceAware="false">

  <Context path="/itcast" docBase="c:\app" />

</Host>
```

虚拟目录名称

Web应用所在目录

在Tomcat6中，不再建议在server.xml文件中配置context元素，细节查看tomcat服务器关于context元素的说明。

让tomcat自动映射：tomcat服务器会自动管理webapps目录下的所有web应用，并把它映射成虚拟目录。换句话说，tomcat服务器webapps目录中的web应用，外界可以直接访问。



# Context元素

- Tomcat6.x提供了多种配置<Context>元素的途径。当其加载一个web应用时，会依次按照以下五种方式尝试查找Web应用的<Context>元素，直到找到为止：
  - 1)到Tomcat安装目录/conf/context.xml文件中查找<Context>元素。
  - 2)到Tomcat安装目录/conf/[enginename]/[hostname]/context.xml.default文件中查找<Context>元素。
- [enginename]: 表示<Engine>的name属性
- [hostname]: 表示<Host>的name属性。
- 3)到Tomcat安装目录/conf/[enginename]/[hostname]/[contextpath].xml文件中查找<Context>元素。
- [contextpath]: 表示单个Web应用的URL入口。

# Context元素

- 4)到Web应用的META-INF/context.xml文件中查找<Context>元素。
- 5) 到Tomcat安装目录/conf/server.xml文件中查找<Context>元素。只适用于单个Web应用。
- 如果仅仅为单个Web应用配置<Context>元素，可以优先采用第三种配置方式。

# Context元素的属性

属性↵	描述↵
<code>path</code> ↵	指定访问该 Web 应用的 URL 入口↵
<code>docBase</code> ↵	指定 Web 应用的文件路径。可以给定绝对路径，也可以给定相对于<Host>的 <code>appBase</code> 属性的相对路径。如果 Web 应用采用开放目录结构，则指定 Web 应用的根目录；如果 Web 应用是个 WAR 文件，则指定 WAR 文件的路径(Tomcat 会自动解压到 <code>webapps</code> 下)。↵
<code>className</code> ↵	指定实现 Context 组件的 Java 类的名字，这个类必须实现 <code>org.apache.catalina.Context</code> 接口，该属性的默认值是 <code>org.apache.catalina.core.StandardContext</code> 。(不建议大家改动)↵
<code>reloadable</code> ↵	如果这个属性值为 <code>true</code> ，Tomcat 服务器在运行状态下会监视在 <code>WEB-INF/classes</code> 和 <code>WEB-INF/lib</code> 目录下的 class 文件的改动，以及监视 Web 应用的 <code>WEB-INF/web.xml</code> 文件的改动，如果检测到有更新，服务器会自动重新加载 web 应用。其默认值是 <code>false</code> 。Web 应用在开放和调试阶段，将其改为 <code>true</code> ，可以方便对 Web 的调试。在正式发布阶段，应将其改为 <code>false</code> ，可以降低 Tomcat 的运行负荷，提高 Tomcat 的运行性能。↵

## 配置虚拟主机

- 假如两家公司Web应用都发布在同一个Tomcat服务器上，可以为每家公司分别创建一个虚拟主机

[www.itcast.cn](http://www.itcast.cn)

[www.itheima.com](http://www.itheima.com)

- 尽管以上两个虚拟主机位于同一个主机，但是当客户通过以上两个不同虚拟主机名访问Web应用时，会感觉到这两个应用分别拥有独立的主机
- 如需在WEB服务器中配置一个网站，需使用Host元素进行配置，例：  
`<Host name="site1" appBase="c:\app"></Host>`
- 配置的主机(网站)要想被外部访问，必须在DNS服务器或windows系统中注册。
- 配置多个网站。
- 缺省虚拟主机。

# Web应用的生命周期

- 用Tomcat的管理平台管理Web应用的生命周期
- Tomcat的管理平台是Tomcat本身的一个Web应用，管理平台对应manager应用，它位于Tomcat安装目录/webapps/manager目录下。
- manager应用会对用户进行安全验证。它要求用户具有manager角色。因此，应该先在Tomcat中添加具有manager角色的用户信息，方法为打开Tomcat安装目录/conf/tomcat-users.xml文件，在其中增加以下内容：
  - <tomcat-users>
  - <role rolename="manager"/>
  - <user username="tomcat" password="tomcat" roles="manager"/>
  - </tomcat-users>



# Thank You!

务实 · 质量 · 创新 · 分享 · 专注 · 责任

[www.itcast.cn](http://www.itcast.cn)