**Tomcat目录结构以及配置文件说明**

**安装包**   
tar.gz文件时Linux操作系统下的安装版本   
exe文件时Windows系统下的安装版本   
zip文件是Windows系统下的压缩版本   
**TOMCAT的目录结构**   
/bin：存放windows或Linux平台上启动和关闭Tomcat的脚本文件   
/conf：存放Tomcat服务器的各种全局配置文件，其中最重要的是server.xml和web.xml   
/doc：存放Tomcat文档   
/server：包含三个子目录：classes、lib和webapps   
/server/lib：存放Tomcat服务器所需的各种JAR文件   
/server/webapps：存放Tomcat自带的两个WEB应用admin应用和 manager应用   
/common/lib：存放Tomcat服务器以及所有web应用都可以访问的jar文件   
/shared/lib：存放所有web应用都可以访问的jar文件（但是不能被Tomcat服务器访问）   
/logs：存放Tomcat执行时的日志文件   
/src：存放Tomcat的源代码   
/webapps：Tomcat的主要Web发布目录，默认情况下把Web应用文件放于此目录   
/work：存放JSP编译后产生的class文件   
WEB应用的目录结构：假设在$CATALINA\_HOME/webapps下有helloapp的web应用   
/helloapp：Web应用的根目录，所有的jsp文件和html文件都在此目录下   
/helloapp/WEB\_INF：存放该web应用发布时的描述文件web.xml   
/helloapp/WEB\_INF/class：存放各种class文件，Servlet文件也存放于此目录下   
/helloapp/WEB\_INF/lib：存放各钟Web应用所需要的jar文件。比如可以存放JDBC驱动程序的JAR文件   
Tomcat加载类和资源的顺序为（以helloapp应用为例）   
helloapp （/webapps/helloapp/Web-INF/下的classes，lib子目录中.calss及.jar，仅helloapp可以加载）   
Bootstrap （$JAVA\_HOME/jre/lib/ext/\*.jar）   
System （$CLASSPATH/\*.class和CLASSPATH中指定的jar）   
Common （$CATALINA\_HOME/common/下的classes，lib，endores子目录中.class及.jar）   
Catalina （$CATALINA\_HOME/server/下的classes，lib子目录中.calss及.jar，仅Tomcat可以加载）   
Shared （$CATALINA\_HOME/shared/下的classes，lib子目录中.calss及.jar，仅所有Web应用可以加载）   
$CATALINA\_HOME/conf/server.xml配置简介   
<**Server**>   
元素属性port：指定一个端口，这个端口负责监听关闭tomcat的请求shutdown：指定向端口发送的命令字符串，通常是“SHUTDOWN”   
<**Service**>   
元素属性name：指定service的名字，通常是”Catalina”   
<**Connector/**>   
元素属性（表示客户端和service之间的连接）：port：指定服务器端要创建的端口号，并使用这个断口监听来自客户端的请求minProcessors（考察是否有此属性）：服务器端启动时创建的处理请求的线程数maxProcessors（考察是否有此属性）：最大可以创建的处理请求的线程数maxThreads：服务器创建的最大线程数minSpareThreads：最小剩余线程数maxSpareThreads：最大剩余线程数enableLookups：如果为true，则可以通过调用request.getRemoteHost（）进行DNS查询来得到远程客户端的实际主机名，若为false则不进行DNS查询，而是返回其IP地址redirectPort：指定当服务器正在处理HTTP请求时收到了一个SSL传输请求后重定向的端口号acceptCount：指定当所有可以使用的处理请求的线程数都被使用时，可以放到处理队列中的请求数，超过的请求将不予处理connectionTimeout：指定超时的时间数（以毫秒为单位）disableUploadTimeout：不明debug：不明protocol：不明   
<**Engine**>   
元素属性（指定service中的请求处理主机，接收和处理来自Connector的请求）name：指定Engine的名字，通常是”Catalina”defaultHost：指定默认的处理请求的主机名，至少与某个host元素的name属性值是相同的   
<**Context/**>   
元素属性（表示一个web应用程序）docBase：指定此Web应用的绝对或相对路径，也可以为WAR文件的路径path：表示此Web应用程序的url的入口，如为“/hello“，则请求的URL为http://localhost:8080/hello/reloadable：如果为true，则Tomcat在运行时会自动监视Web应用的/WEB-INF/lib和/WEB-INF/classes下文件的改动，自动装载新应用，使我们可以在不重启Tomcat的情况下更新Web应用   
<**Host**>   
元素属性（表示一个虚拟主机）name：指定虚拟主机名字debug：指定日志级别appBase：存放Web应用程序的基本目录，可以是绝对路径或相对于$CATALINA\_HOME的目录，默认是$CATALINA\_HOME/webappsunpackWARs：如果为true，则tomcat会自动将WAR文件解压后运行，否则不解压而直接从WAR文件中运行应用程序autoDeploy：如果为true，表示Tomcat启动时会自动发布appBase目录下所有的Web应用（包括新加入的Web应用）deployOnStarup：如果此项为true，表示Tomcat服务器启动时会自动发布appBase目录下所有Web应用。如果Web应用在server.xml中没有相应的元素,则将采用默认的Context配置。deployOnStarup的默认设置是truealias：指定虚拟主机的别名，可以指定多个别名xmlValidation：不明xmlNamespaceAware：不明   
<**Logger/**>   
元素属性（表示日志，调试和错误信息）className：指定logger使用的类名，此类必须实现org.apache.catalina.Logger 接口prefix：指定log文件名的前缀（文件名）suffix：指定log文件名的后缀（扩展名）timestamp：如果为true，则log文件名中会加入日期时间，如下例：localhost\_log.2006-10-04.txtdirectory：指定log文件存放的目录   
<**Realm/**>   
元素属性（表示存放用户名，密码及role的数据库）className：指定Realm使用的类名，此类必须实现org.apache.catalina.Realm接口resourceName：不明   
<**Valve/**>   
元素属性（功能与Logger相似，其prefix和suffix属性解释和Logger 中的一样）className：指定Valve使用的类名，如用org.apache.catalina.valves.AccessLogValve类可以记录应用程序的访问信息directory：指定log文件存放的位置pattern：有两个值，common方式记录远程主机名或ip地址、用户名、日期、第一行请求的字符串、HTTP响应代码、发送的字节数。combined方式比common方式记录的值还多   
$CATALINA\_HOME/conf/web.xml配置简介   
默认Web应用的首页文件的设置

<welcome-file-list>

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

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

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

</welcome-file-list>

报错文件的设置   
a.如果某文件资源没有找到，服务器要报404错误，按下述配置则会调用\webapps\ROOT\notFileFound.jsp

<error-page>

<error-code>404</error-code>

<location>/notFileFound.jsp</location>

</error-page>

b.如果执行的某个JSP文件产生NullPointException，则会调用\webapps\ROOT\null.jsp

<error-page>

<exception-type>java.lang.NullPointerException</exception-type>

<location>/null.jsp</location>

</error-page>

session 会话过期时间的设置,单位是分钟

<session-config>

<session-timeout>30</session-timeout>

</session-config>

**servlet的设置**

<servlet>

<servlet-name>default</servlet-name>

<servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>

<init-param>

<param-name>debug</param-name>

<param-value>0</param-value>

</init-param>

<init-param>

<param-name>listings</param-name>

<param-value>true</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

说明：：定义Servlet的名字。：指定实现这个Servlet的类。：定义Servlet的初始化参数（包括参数名和参数值），一个 元素中可以有多个元素。：指定当Web应用启动时，装载Servlet的次序。当这个值为正数或零，Servlet容器先加载数值小的Servlet，再一次加载其他数值大的Servlet。如果这个值为负数或没有设定，Servlet将在Web客户首次访问这个Servlet时加载它。   
filter过滤器的设置

<filter>

<filter-name>FilterSource</filter-name>

<filter-class>project4. FilterSource </filter-class>

</filter>

<filter-mapping>

<filter-name>FilterSource</filter-name>

<url-pattern>/WwwServlet</url-pattern>

(<url-pattern>/haha/\*</url-pattern>)

</filter-mapping>

过滤：1) 身份验证的过滤Authentication Filters2) 日志和审核的过滤Logging and Auditing Filters3) 图片转化的过滤Image conversion Filters4) 数据压缩的过滤Data compression Filters5) 加密过滤Encryption Filters6) Tokenizing Filters7) 资源访问事件触发的过滤Filters that trigger resource access events XSL/T 过滤XSL/T filters9) 内容类型的过滤Mime-type chain Filter注意监听器的顺序，如：先安全过滤，然后资源，然后内容类型等，顺序可以自己定。   
Tomcat管理   
管理员帐户配置（$CATALINA\_HOME/conf/tomcat-users.xml）在tomcat-users.xml文件相应的位置添加一行：<user name="user" password="user" roles="standard,manager"/>用来添加一个管理员帐户。然后重起tomcat，在浏览器中输入地址http://localhost:8080/manager/，在弹出的对话框输入上面的用户名和密码即可   
Web应用列表在浏览器中输入http://localhost:8080/manager/list，浏览器将会列出目前存在的所有Web应用的信息，分别为：Web应用程序的路径、当前状态、连接这个Web应用的Session数目   
重新装载Web应用程序在浏览器中输入 http://localhost:8080/manager/reload?path=/examples，浏览器显示如下：OK - Reloaded application at context path /examples，这表示example应用程序装载成功，如果我们将server.xml的Context元素的reloadable属性设为true，则没必要利用这种方式重新装载Web应用程序，因为tomcat会自动装载。   
显示Session信息在浏览器中输入http://localhost:8080/manager/sessions?path=/examples，浏览器显示如下：OK - Session information for application at context path /examples Default maximum session inactive interval 30 minutes   
启动和关闭应用程序在浏览器中输入http://localhost:8080/manager/start?path=/example和http://localhost:8080/manager/stop?path=/examples分别启动和关闭examples应用程序。   
部署及撤销部署WAR有两种组织方式，一种是按一定的目录结构组织文件，一种是一个后缀为WAR的压缩包，因此它的部署方式也有两种：a.在浏览器中输入：http://localhost:8080/manager/install?path=/examples&war=file:/c:\examples，就会将按目录结构组织的WAR部署b.在浏览器中输入：http://localhost:8080/manager/install?path=/examples&war=jar:file:/c:\examples.war!/，就会将按压缩包组织的WAR部署，注意此url后半部分一定要有!/号部署后就可以用：http://localhost:8080/examples访问了在浏览器中输入：http://localhost:8080/manager/remove?path=/examples，就会撤销刚才部署的应用程序

**更多Tomcat相关教程见以下内容**：

[**CentOS**](http://www.linuxidc.com/topicnews.aspx?tid=14)**6.6下安装配置Tomcat环境**  <http://www.linuxidc.com/Linux/2015-08/122234.htm>

[RedHat](http://www.linuxidc.com/topicnews.aspx?tid=10) Linux 5.5安装JDK+Tomcat并部署Java项目  <http://www.linuxidc.com/Linux/2015-02/113528.htm>

Tomcat权威指南(第二版)(中英高清PDF版+带书签)  <http://www.linuxidc.com/Linux/2015-02/113062.htm>

Tomcat 安全配置与性能优化 <http://www.linuxidc.com/Linux/2015-02/113060.htm>

Linux下使用Xshell查看Tomcat实时日志中文乱码解决方案 <http://www.linuxidc.com/Linux/2015-01/112395.htm>

CentOS 64-bit下安装JDK和Tomcat并设置Tomcat开机启动操作步骤 <http://www.linuxidc.com/Linux/2015-01/111485.htm>

CentOS 6.5下安装Tomcat  <http://www.linuxidc.com/Linux/2015-01/111415.htm>

Tomcat中session的管理机制  <http://www.linuxidc.com/Linux/2016-09/135072.htm>

**Tomcat 的详细介绍**：[请点这里](http://www.linuxidc.com/Linux/2013-10/91681.htm)  
**Tomcat 的下载地址**：[请点这里](http://www.linuxidc.com/down.aspx?id=1058)

**本文永久更新链接地址**：<http://www.linuxidc.com/Linux/2016-10/135783.htm>