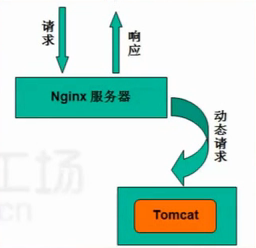
# Tomcat安装教程

## 简介

Tomcat 类似与一个 apache 的扩展型，属于 apache 软件基金会的核心项目，属于开源的轻量级 Web 应用服务器，是开发和调试 JSP 程序的首选，主要针对 Jave 语言开发的网页代码进行解析，Tomcat 虽然和 Apache 或者 Nginx 这些 Web 服务器一样，具有处理 HTML 页面的功能，然而由于其处理静态 HTML 的能力远不及 Apache 或者 Nginx，所以 Tomcat 通常做为一个 Servlet 和 JSP 容器单独运行在后端。可以这样认为，当配置正确时，Apache 为 HTML 页面服务，而 Tomcat 实际上运行 JSP 页面和 Servlet。比如 apache 可以通过 cgi 接口直接调取 Tomcat 中的程序。



## 准备

由于Tomcat是由JAVA语言写的，所以需要先安装Jdk配置JAVA环境后才能运行Tomcat。

## 配置内容说明

1. 主目录结构

\bin 存放启动和关闭 Tomcat 的可执行文件

\conf 存放 Tomcat 的配置文件

\lib 存放库文件

\logs 存放日志文件

\temp 存放临时文件

\webapps 存放 web 应用，部署的应用都在这里

\work 存放 JSP 转换后的 Servlet 文件

1. 配置文件

catalina.policy：权限控制配置文件

catalina.properties：Tomcat属性配置文件

context.xml：上下文配置文件

logging.properties：日志log相关配置文件

server.xml：主配置文件

Tomcat-users.xml：manager-gui管理用户配置文件

web.xml：Tomcat的servlet、servlet-mapping、filter、MIME等相关配置

1. server.xml主配置文件（描述了如何启动Tomcat）

<Server>

<Listener />

<GlobaNamingResources>

</GlobaNamingResources

<Service>

<Connector />

<Engine>

<Logger />

<Realm />

<host>

<Logger />

<Context />

</host>

</Engine>

</Service>

</Server>

|  |  |  |
| --- | --- | --- |
| 元素名 | 属性 | 解释 |
| server | 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 |
| service | name | 指定 service 的名字 |
| Connector(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，并在这个断口监听来自客户端的请求 |
| minProcessors | 服务器启动时创建的处理请求的线程数 |
| maxProcessors | 最大可以创建的处理请求的线程数 |
| 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，超过这个数的请求将不予处理 |
| connectionTimeout | 指定超时的时间数 (以毫秒为单位) |
| Engine(表示指定 service 中的请求处理机，接收和处理来自 Connector 的请求) | defaultHost | 指定缺省的处理请求的主机名，它至少与其中的一个 host 元素的 name 属性值是一样的 |
| Context(表示一个 web 应用程序，通常为 WAR 文件，关于 WAR 的具体信息见 servlet 规范) | docBase | 应用程序的路径或者是 WAR 文件存放的路径 |
| path | 表示此 web 应用程序的 url 的前缀，这样请求的 url 为 http://localhost:8080/path/\*\*\*\* |
| reloadable | 这个属性非常重要，如果为 true，则 tomcat 会自动检测应用程序的 / WEB-INF/lib 和 / WEB-INF/classes 目录的变化，自动装载新的应用程序，我们可以在不重起 tomcat 的情况下改变应用程序 |
| host(表示一个虚拟主机) | name | 指定主机名 |
| appBase | 应用程序基本目录，即存放应用程序的目录 |
| unpackWARs | 如果为 true，则 tomcat 会自动将 WAR 文件解压，否则不解压，直接从 WAR 文件中运行应用程序 |
| Logger(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 文件的前缀 |
| suffix | 指定 log 文件的后缀 |
| timestamp | 如果为 true，则 log 文件名中要加入时间，如下例: localhost\_log.001-10-04.txt |
| Realm(表示存放用户名，密码及 role 的**[数据库](http://lib.csdn.net/base/14" \o "MySQL知识库" \t "_blank)**) | className | 指定 Realm 使用的类名，此类必须实现 org.apache.catalina.Realm 接口 |
| Valve(功能与 Logger 差不多，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 方式记录的值更多 |

# 其他

## localhost:8080访问主页原理

1. 解析域名localhost到127.0.0.1，端口为server,xml设置的8080
2. 访问webapps文件夹，如果端口后带有项目则访问具体项目，如果没有项目会默认访问ROOT项目（文件夹）。
3. 主页就是ROOT项目

## 启动tomcat的方式

1. 直接使用tomcat自带的bat程序启动
2. Cmd中进入到tomcat安装目录的bin文件夹下，使用命令catalina进行启动