

tomcat server.xml 配置详解

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，并在这个端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，我们可以在不重启 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.2001-10-04.txt
Realm (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
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

<Server>元素

它代表整个容器, 是 Tomcat 实例的顶层元素. 由 org.apache.catalina.Server 接口来定义. 它包含一个 <Service>元素. 并且它不能做为任何元素的子元素.

```
<Server port="8005" shutdown="SHUTDOWN" debug="0">
```

- 1> className 指定实现 org.apache.catalina.Server 接口的类. 默认值为 org.apache.catalina.core.StandardServer
- 2> port 指定 Tomcat 监听 shutdown 命令端口. 终止服务器运行时, 必须在 Tomcat 服务器所在的机器上发出 shutdown 命令. 该属性是必须的.
- 3> shutdown 指定终止 Tomcat 服务器运行时, 发给 Tomcat 服务器的 shutdown 监听端口的字符串. 该属性必须设置

<Service>元素

该元素由 org.apache.catalina.Service 接口定义, 它包含一个 <Engine>元素, 以及一个或多个 <Connector>, 这些 Connector 元素共享用同一个 Engine 元素

```
<Service name="Catalina">  
<Service name="Apache">
```

第一个<Service>处理所有直接由 Tomcat 服务器接收的 web 客户请求.

第二个<Service>处理所有由 Apache 服务器转发过来的 Web 客户

- 1> className 指定实现 org.apache.catalina.Service 接口的类. 默认为 org.apache.catalina.core.StandardService
- 2> name 定义 Service 的名字

<Engine>元素

每个 Service 元素只能有一个 Engine 元素. 元素处理在同一个<Service>中所有<Connector>元素接收到的客

户请求. 由 org.apache.catalina.Engine 接口定义.

```
<Engine name="Catalina" defaultHost="localhost" debug="0">
```

- 1> className 指定实现 Engine 接口的类, 默认值为 StandardEngine
 - 2> defaultHost 指定处理客户的默认主机名, 在<Engine>中的<Host>子元素中必须定义这一主机
 - 3> name 定义 Engine 的名字
- 在<Engine>可以包含如下元素<Logger>, <Realm>, <Value>, <Host>

<Host>元素

它由 Host 接口定义. 一个 Engine 元素可以包含多个<Host>元素. 每个<Host>的元素定义了一个虚拟主机. 它包含了一个或多个 Web 应用.

```
<Host name="localhost" debug="0"  
appBase="webapps" unpackWARs="true" autoDeploy="true">
```

- 1> className 指定实现 Host 接口的类. 默认值为 StandardHost
- 2> appBase 指定虚拟主机的目录, 可以指定绝对目录, 也可以指定相对于<CATALINA_HOME>的相对目录. 如果没有此项, 默认为<CATALINA_HOME>/webapps
- 3> autoDeploy 如果此项设为 true, 表示 Tomcat 服务处于运行状态时, 能够监测 appBase 下的文件, 如果有新有 web 应用加入进来, 会自动发布这个 WEB 应用
- 4> unpackWARs 如果此项设置为 true, 表示把 WEB 应用的 WAR 文件先展开为开放目录结构后再运行. 如果设为 false 将直接运行为 WAR 文件
- 5> alias 指定主机别名, 可以指定多个别名
- 6> deployOnStartup 如果此项设为 true, 表示 Tomcat 服务器启动时会自动发布 appBase 目录下所有的 Web 应用. 如果 Web 应用中的 server.xml 没有相应的<Context>元素, 将采用 Tomcat 默认的 Context

7> name 定义虚拟主机的名字

在<Host>元素中可以包含如下子元素

<Logger>, <Realm>, <Value>, <Context>

<Context>元素

它由 Context 接口定义. 是使用最频繁的元素. 每个<Context 元素代表了运行在虚拟主机上的单个 Web 应用.

一个<Host>可以包含多个<Context>元素. 每个 web 应用有唯一

的一个相对应的 Context 代表 web 应用自身. servlet 容器为第一个 web 应用创建一个

ServletContext 对象.

```
<Context path="/sample" docBase="sample" debug="0" reloadbale="true">
```

1> className 指定实现 Context 的类, 默认为 StandardContext 类

2> path 指定访问 Web 应用的 URL 入口, 注意/myweb, 而不是 myweb 了事

3> reloadable 如果这个属性设为 true, Tomcat 服务器在运行状态下会监视在 WEB-INF/classes 和 Web-

INF/lib 目录 CLASS 文件的改运. 如果监视到有 class 文件被更新, 服务器自重新加载 Web 应用

3> cookies 指定是否通过 Cookies 来支持 Session, 默认值为 true

4>useNaming 指定是否支持 JNDI, 默认值为了 true

在<Context>元素中可以包含如下元素

<Logger>, <Realm>, <Resource>, <ResourceParams>

<Connector>元素

由 Connector 接口定义. <Connector>元素代表与客户程序实际交互的给件, 它负责接收客户请求, 以及向客户返回响应结果.

```
<Connector port="8080" maxThread="50" minSpareThreads="25"
maxSpareThread="75" enableLookups="false"
redirectPort="8443" acceptCount="100" debug="0"
connectionTimeout="20000" disableUploadTimeout="true" />
<Connection port="8009" enableLookups="false" redirectPort="8443"
debug="0" protocol="AJP/1.3" />
```

第一个 Connector 元素定义了一个 HTTP Connector, 它通过 8080 端口接收 HTTP 请求;第二个 Connector 元素定

义了一个 JD Connector, 它通过 8009 端口接收由其它服务器转发过来的请求.

Connector 元素共用属性

1> className 指定实现 Connector 接口的类

2> enableLookups 如果设为 true, 表示支持域名解析, 可以把 IP 地址解析为主机名. WEB 应用中调用

request.getRemoteHost 方法返回客户机主机名. 默认值为 true

3> redirectPort 指定转发端口. 如果当前端口只支持 non-SSL 请求, 在需要安全通信的场合, 将把客户请求转

发至 SSL 的 redirectPort 端口

HttpConnector 元素的属性

1> className 实现 Connector 的类

2> port 设定 Tcp/IP 端口, 默认值为 8080, 如果把 8080 改成 80, 则只要输入

<http://localhost> 即可

因为 TCP/IP 的默认端口是 80

3> address 如果服务器有二个以上 ip 地址, 此属性可以设定端口监听的 ip 地址.

默认情况下, 端口会监听服务

器上所有的 ip 地址

4> bufferSize 设定由端口创建的输入流的缓存大小. 默认值为 2048byte

5> protocol 设定 Http 协议, 默认值为 HTTP/1.1

6> maxThreads 设定在监听端口的线程的最大数目, 这个值也决定了服务器可以同时响应客户请求的最大数

目. 默认值为 200

7> acceptCount 设定在监听端口队列的最大客户请求数量, 默认值为 10. 如果队列已满, 客户必须等待.

8> connectionTimeout 定义建立客户连接超时的时间. 如果为-1, 表示不限制建立客户连接的时间

JkConnector 的属性

1> className 实现 Connector 的类

2> port 设定 AJP 端口号

3> protocol 必须设定为 AJP/1.3