

# Tomcat 学习笔记

## Tomcat 是什么?

Tomcat 服务器 Apache 软件基金会项目中的一个核心项目，是一个免费的**开放源代码的 Web 应用服务器**，属于**轻量级应用服务器**，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。

## Connector 运行模式

**BIO**: 同步并阻塞 一个线程处理一个请求。缺点：并发量高时，线程数较多，浪费资源。Tomcat7 或以下，在 Linux 系统中默认使用这种方式。

**NIO**: 同步非阻塞 IO

利用 Java 的异步 IO 处理，可以通过少量的线程处理大量的请求，可以复用同一个线程处理多个 connection(多路复用)。

**APR**: 即 Apache Portable Runtime，从操作系统层面解决 io 阻塞问题。**\*\*AIO 方式，\*\***异步非阻塞 IO(Java NIO2 又叫 AIO) 主要与 NIO 的区别主要是操作系统的底层区别.可以做个比喻:比作快递，NIO 就是网购后要自己到官网查下快递是否已经到了(可能是多次)，然后自己去取快递；AIO 就是快递员送货上门了(不用关注快递进度)。

## 几种部署方式

1. 利用 Tomcat 的自动部署。

把 web 应用拷贝到 **webapps** 目录。Tomcat 在启动时会加载目录下的应用，并将编译后的结果放入 work 目录下。

2. 使用 Manager App 控制台部署。

在 tomcat 主页点击“Manager App”进入应用管理控制台，可以指定一个 web 应用的路径或 war 文件。

3. 修改 conf/server.xml 文件部署。

修改 conf/server.xml 文件，增加 Context 节点可以部署应用。

4. 增加自定义的 Web 部署文件。

在 conf/Catalina/localhost/ 路径下增加 xyz.xml 文件，内容是 Context 节点，可以部署应用。

## 如何创建 servlet 类实例

1. 当容器启动时，会读取在 webapps 目录下所有的 web 应用中的 web.xml 文件，然后对 xml 文件进行解析，并读取 **servlet** 注册信息。然后，将每个应用中注册的 **servlet** 类都进行加

载，并通过 反射的方式实例化。（有时候也是在第一次请求时实例化）

2. 在 `servlet` 注册时加上 1 如果为正数，则在一开始就实例化，如果不写或为负数，则第一次请求实例化。

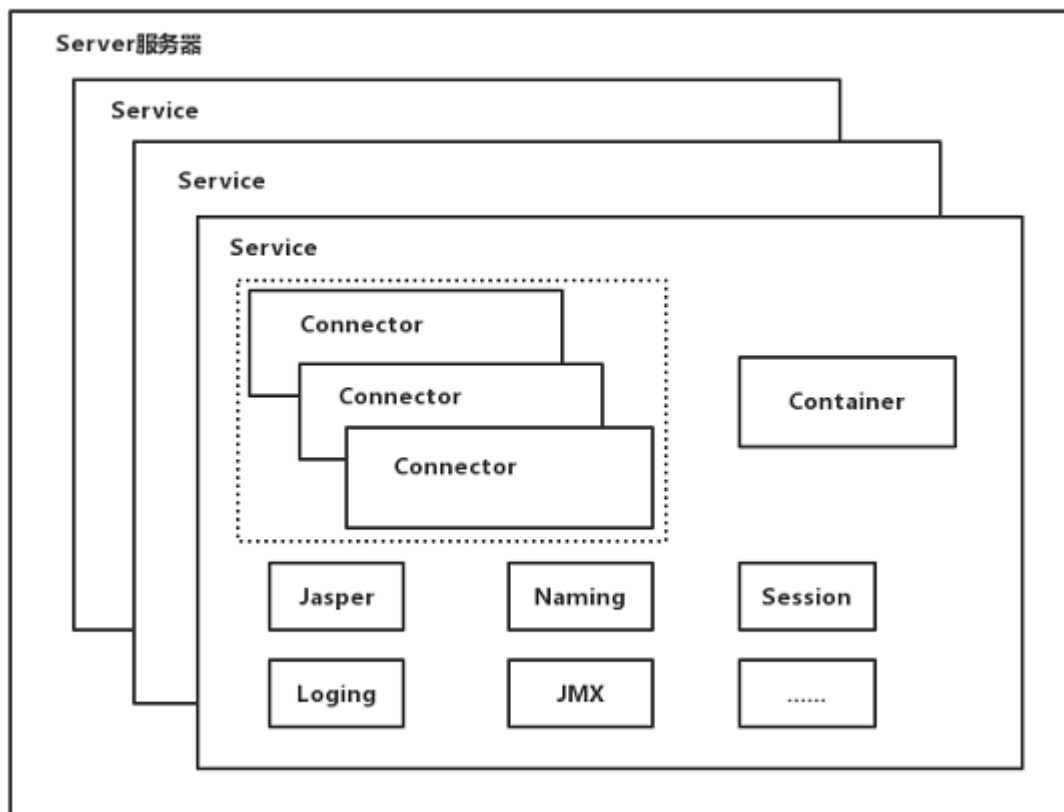
## Tomcat 工作模式

Tomcat 作为 `servlet` 容器，有三种工作模式：

- 1、独立的 `servlet` 容器，`servlet` 容器是 web 服务器的一部分；
  - 2、进程内的 `servlet` 容器，`servlet` 容器是作为 web 服务器的插件和 `java` 容器的实现，web 服务器插件在内部地址空间打开一个 `jvm` 使得 `java` 容器在内部得以运行。反应速度快但伸缩性不足；
  - 3、进程外的 `servlet` 容器，`servlet` 容器运行于 web 服务器之外的地址空间，并作为 web 服务器的插件和 `java` 容器实现的结合。反应时间不如进程内但伸缩性和稳定性比进程内优；
- 进入 Tomcat 的请求可以根据 Tomcat 的工作模式分为如下两类：

Tomcat 作为应用程序服务器：请求来自于前端的 web 服务器，这可能是 Apache，IIS，Nginx 等；  
Tomcat 作为独立服务器：请求来自于 web 浏览器；

## Tomcat 顶层架构



<http://blog.csdn.net/xlgen157387>

<https://blog.csdn.net/ThinkWon/article/details/104397665>