

**Course Length:** 2 days

**Course Description:** Apache Tomcat is the most popular platform for deploying Java-based™ Web applications. In this two days course, attendees learn how to administer the Tomcat server, deploy applications on the server, ensure the server's security, troubleshoot problems, and cluster Tomcat to ensure high availability.

**Prerequisites:** All attendees should be familiar with general principles of Web server administration and have some experience building Web applications. Prior experience with Java as an application server administrator or developer is helpful but not required.

**Detailed description:**

Introduction

- Overview of the Apache Software Foundation and the Jakarta Project
- Overview of Java EE as a platform
- Overview of the features and functionality specifically provided by Tomcat
- Installing Tomcat
  - Installing the Java Runtime Environment (JRE)
  - Tweaking the JRE for performance
  - Performing the actual Tomcat installation
- Examining the Tomcat installation directories
  - bin
  - conf
  - common (Tomcat 5.5 and earlier)
  - lib (Tomcat 6 and later)
  - logs
  - server (Tomcat 5.5 and earlier)
  - shared (Tomcat 5.5 and earlier)
  - temp
  - webapps
  - work
- Configuring Tomcat
  - server.xml (detailed walkthrough)
  - web.xml
  - context.xml
- Tomcat Valves
  - AccessLog and FastCommonAccessLog
  - RequestFilterValve
  - SingleSignOnValve (by request)
  - RequestDumperValve (by request)
- Memory management and JMX monitoring
  - Understanding Java garbage collection
  - Using JAVA\_OPTS, JMX and JConsole to monitor and tune Tomcat memory usage
  - Sizing Tomcat's JVM memory heap
  - Using JMX and JConsole to configure Tomcat via Tomcat's MBeans
  - Updating Tomcat's configuration via JMX "on the fly" without restarting Tomcat
  - Load testing with JMeter

- Using VisualVM (new monitoring tool built into JDK 6) and PSI Probe
  - Controlling JMX MBeans via Ant
- Logging
  - JULI logging
  - log4j logging
  - Understanding exceptions and thread dumps
- Connecting databases with Tomcat applications
  - Classic JDBC approach
  - Better approach: JNDI resources
  - Setting up and monitoring database connection pools
- Security
  - File system security
  - Java security manager
  - Realms, authentication, and authorization
  - SSL
- Performance tuning strategies
  - Additional JVM tuning tips
  - Enabling parallel garbage collection
  - Building native connectors
  - Disabling/removing unneeded applications
  - Tuning incoming connections and database connection pools
  - Turning off Jasper development mode
  - Precompiling JSPs
  - Preloading servlets
- Tomcat 7 New Features [optional]
  - Memory leak prevention and detection
  - Servlet 3.0, JSP 2.2, and EL 2.2 support (and the practical implications of this for Tomcat admins)
  - Cross-site request forgery prevention (and how to configure)
  - How Tomcat 7's new session management features prevent session fixation attacks
  - Alias support (which allow static content to be stored outside the WAR file)
- Running Tomcat behind Apache httpd or IIS [this section would be taught using your web server and connector module of choice]
  - Why run Tomcat behind Apache httpd or IIS?
  - Installing mod\_jk (Apache or Tomcat) or mod\_proxy\_ajp and mod\_proxy\_balancer (Apache 2.2 or later only)
  - Proxying traffic to Tomcat via AJP
  - Monitoring the status of your web server's connection to Tomcat
  - Load balancing Tomcat via mod\_jk or mod\_proxy\_balancer
- Tomcat Clustering
  - Configuring mod\_jk (in Apache or IIS) or mod\_proxy\_balancer (Apache 2.2 or later only) as a load balancer
  - Hardware load balancing as an alternative to software load balancing
  - Sticky sessions
  - Configuring a shared session back-end
    - FileStore/JDBCStore (older approach, generally not recommended)
    - Using the <Cluster /> tag
  - Configuring the application to be distributable
  - Setting up and testing failover