

Tomcat Server Administration

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, how to deploy applications on the server, and how to ensure the server's security.

Group rates available – Inquire for more details

Duration: 24 hours

Course Overview

- To learn how to setup and configure Apache Tomcat
- To master deploying Java web applications to the Tomcat server
- To learn how to configure Tomcat valves for access logging, single sign-on, and access control
- To discover how to secure applications running on the Tomcat server
- To explore using Ant for deploying applications to the Tomcat server
- To master configuring Tomcat to run behind Apache using the mod_jk modul

Prerequisites

- All attendees should be familiar with general principles of Web server administration and have some experience building Web applications.

Hands-on/Lecture Ratio

This Tomcat training class is 70% hands-on, 30% lecture, with the longest lecture segments lasting for 20 minutes. Students "learn by doing," with immediate opportunities to apply the material they learn to real-world problems.

Course Content

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

Tomcat sessions and persistence

- Standard session behavior
- Disabling session saves across restarts (if desired)
- FileStore
- JDBCStore
- Using <Cluster /> to replicate sessions across Tomcat instances

Connecting databases with Tomcat applications

- Classic JDBC approach
- Better approach: JNDI resources

Security

- File system security
- Java security manager
- Realms
 - Using the UserDatabaseRealm
 - Using the JDBC realm
- 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 with Apache

- Introducing the mod_jk connector [mod_proxy_ajp taught in lieu of mod_jk upon request]
- Understanding Tomcat workers
- Connecting Tomcat with Apache
- Configuring SSL
- Tomcat load balancing with Apache

Tomcat 5 clustering

- Clustering model
 - Load balancing
 - Session sharing
- Working with Tomcat clustering
 - Session management
 - Configuring a cluster
 - Testing the cluster