

**Administering Apache Tomcat**

**Course Number:** SRV-100
**Duration:** 2 days

**Overview**

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

This course can be taught using Tomcat 11.0, 10.1, 10.0, 9.0, 8.5, or any other version, with topics added or removed as appropriate to your version. This course can also be adapted to teach administration of TomEE Web Profile.

**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.

**Materials**

All attendees receive Accelebrate's comprehensive courseware and a two-page checklist of performance tuning and security suggestions.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computer for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine.

**Objectives**

* Set up and configure Apache Tomcat
* Deploy Java web applications to the Tomcat server
* Configure Tomcat valves for access logging, single sign-on, and access control
* Monitor Tomcat via its JMX MBeans and a variety of tools, including JConsole, VisualVM, and PSI Probe
* Tune Tomcat for optimal performance
* Configure Tomcat logs and troubleshoot Tomcat
* Secure Tomcat
* Build and monitor database connection pools
* Run Tomcat behind a web server, such as Apache httpd or Microsoft IIS
* Build Tomcat clusters to ensure high availability

**Outline**

* Introduction
	+ Overview of the Apache Software Foundation
	+ Overview of Jakarta EE (formerly Java EE) as a platform
	+ Overview of the features and functionality specifically provided by Tomcat
* Installing Tomcat
	+ Installing the Java Runtime Environment (JRE)
	+ Performing the Tomcat installation (with nuances specific to your operating system discussed)
* Examining the Tomcat installation directories
	+ bin
	+ conf
	+ lib
	+ logs
	+ temp
	+ webapps
	+ work
* Configuring Tomcat
	+ server.xml (detailed walkthrough)
	+ web.xml
	+ context.xml
* Tomcat Valves
	+ AccessLog
	+ RequestFilterValve
	+ Selective coverage of other valves and/or filters as needed in your environment
	+ Discussion of how filters are progressively replacing valves
* 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
	+ Using VisualVM and PSI Probe
	+ Commercial monitoring alternatives
* Logging
	+ JULI logging
	+ Configuring handlers and sources
	+ 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 (if appropriate in your environment)
	+ SSL (optional)
	+ Closing potential security holes in Tomcat’s default configuration
		- Turning off the shutdown port
		- Ensuring directory listings are disabled
		- Removing unnecessary applications
		- Password protecting the JMX port (if open)
		- Turning off unnecessary connectors
		- Hardening the remaining connectors
		- Hiding the version of Tomcat you are running
		- Additional hardening strategies
* Performance tuning strategies
	+ Additional JVM tuning tips
	+ Changing to a different garbage collector
	+ Choosing the right connector (NIO or APR)
	+ Disabling/removing unneeded applications
	+ Tuning incoming connections and back-end database connection pools
	+ Turning off Jasper development mode
	+ Precompiling JSPs
	+ Preloading servlets
	+ Turning off autodeploy and automatic watching of web.xml files
	+ Memory leak prevention and detection
	+ Alias support (which allow static content to be stored outside the WAR file)
	+ URL rewriting
* 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?
	+ mod\_proxy\_ajp and mod\_proxy\_balancer (Apache httpd), or URL Rewrite and ARR (IIS only)
	+ Proxying traffic to Tomcat via AJP or HTTP
	+ Monitoring the status of your web server's connection to Tomcat
* Tomcat Clustering
	+ Configuring mod\_proxy\_balancer (Apache httpd) or ARR (IIS) as a load balancer
	+ Hardware load balancing as an alternative to software load balancing
	+ Sticky sessions
	+ Session Replication using <Cluster />
	+ Configuring the application to be distributable
	+ Setting up and testing failover
	+ Alternative session replication back-ends
* Conclusion