Software Platforms

LM in Computer Engineering

Massimo Maresca

Apache Tomcat

per avviare tomcat: scaricalo aprilo vai nella cartella bin e setta variabili d'ambiente poi scrivi startup

- One of the best known and most widely used Servlet Containers
- Works both as a Web Server and as a Servlet Container
- Open Source
- De Facto Standard

- First Activity: Download and Install
- Second Activity: Look at the documentation

Directories

These are some of the key tomcat directories:

- /bin - Startup, shutdown, and other scripts. The *.sh files (for Unix systems) are functional duplicates of the *.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.
- /conf - Configuration files and related DTDs. The most important file in here is server.xml. It is the main configuration file for the container.
- /logs - Log files are here by default.
- /webapps - This is where your webapps go.

CATALINA_HOME and CATALINA_BASE

Throughout the documentation, there are references to the two following properties:

- CATALINA_HOME: Represents the root of your Tomcat installation, for example /home/tomcat/apache-tomcat-9.0.10 or C:\Program Files\apache-tomcat-9.0.10.
- CATALINA_BASE: Represents the root of a runtime configuration of a specific Tomcat instance. If you want to have multiple Tomcat instances on one machine, use the CATALINA_BASE property.
- By default, CATALINA_HOME and CATALINA_BASE point to the same directory. Set CATALINA_BASE manually when you require running multiple Tomcat instances on one machine.

Tomcat Activation

- Download from Apache
- Environment Variable configuration:

  JAVA_HOME

  TOMCAT_HOME

  CATALINA_BASE , CATALINA_HOME

- The /bin directory

  Main commands: catalina start (startup) stdout and errors on the log file (TOMCAT_HOME/logs/catalina.out

  catalina stop (shutdown)

Management

- Management is done through a servlet. Of course the servlet can access the container under an authorization scheme.
- Authorizations:

  It is necessary to edit file `tomcat-users.xml` located in the `conf` directory

```
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
```

- For example add the following line:

```
<user username="max" password="max" roles="manager-gui"/>
```

A Web Application (Chapter 9 of the Servlet API Specification, version 2.3)

- A Web Application is defined as a hierarchy of directories and files in a standard layout. Such a hierarchy can be accessed in its "unpacked" form, where each directory and file exists in the filesystem separately, or in a "packed" form known as a Web ARchive, or WAR file. The former format is more useful during development, while the latter is used when you distribute your application to be installed.
- The top-level directory of your web application hierarchy is also the document root of your application. Here, you will place the HTML files and JSP pages that comprise your application's user interface. When the system administrator deploys your application into a particular server, he or she assigns a context path to your application. Thus, if the system administrator assigns your application to the context path /catalog, then a request URI referring to /catalog/index.html will retrieve the index.html file from your document root.

Directory Organization (1)

To facilitate creation of a Web Application Archive file in the required format, it is convenient to arrange the "executable" files in the same organization as required by the WAR format itself. To do this, you will end up with the following contents in your application's "document root" directory:

- *.html, *.jsp, etc. - The HTML and JSP pages, along with other files that must be visible to the client browser (such as JavaScript, stylesheet files, and images.
- /WEB-INF/web.xml - The Web Application Deployment Descriptor for your application. This is an XML file describing the servlets and other components that make up your application, along with any initialization parameters and container-managed security constraints that you want the server to enforce for you.
- A /META-INF/context.xml file can be used to define Tomcat specific configuration options, such as an access log, data sources, session manager configuration and more.

Directory Organization (2)

- /WEB-INF/classes/ - This directory contains any Java class files (and associated resources) required for your application, including both servlet and non-servlet classes, that are not combined into JAR files. If your classes are organized into Java packages, you must reflect this in the directory hierarchy under /WEB-INF/classes/.
- /WEB-INF/lib/ - This directory contains JAR files that contain Java class files (and associated resources) required for your application.
- When you install an application into Tomcat, the classes in the WEB-INF/classes/ directory, as well as all classes in JAR files found in the WEB-INF/lib/ directory, are made visible to other classes within your particular web application.

Deployment

A web application can be deployed in Tomcat by one of the following approaches:

- Copy unpacked directory hierarchy into a subdirectory in directory $CATALINA_BASE/webapps/. Tomcat will assign a context path to your application based on the subdirectory name you choose.
- Copy the web application archive file into directory $CATALINA_BASE/webapps/. Tomcat automatically expands the web application archive file into its unpacked form and execute the application that way.
- Use the Tomcat "Manager" web application to deploy and undeploy web applications. Tomcat includes a web application, deployed by default on context path /manager, that allows you to deploy and undeploy applications on a running Tomcat server without restarting it.

Analysis (1)

- Analysis of a simple servlet implementing the doGet method, built through CLI commands.

  Folder: `Basic-Tomcat-Servlet-get`

- Analysis of a simple servlet implementing the doPost method, built through CLI commands.

  Folder: `Basic-Tomcat-Servlet-POST-with-Form-Access`

Folder Analysis, Compilation, Deployment, Execution

Analysis on Google Chrome with Develeopment Tools: Ctrl-SHFT J

Analysis (2)

- Creation of a simple servlet implementing the doGet method and the doPOST method, built using Eclipse.

Deployment and Execution
within local Eclipse Tomcat Server

vs.

Exportation as a WAR file for deployment and execution
within the regular Tomcat server