CHAPTER 4

Using Tomcat’s Manager Web Application

In previous chapters, we have learned, among other things, how to deploy Java web applications to Apache Tomcat server. In our examples, we copied the directory or WAR file of the web application to Tomcat’s /webapps directory, and started (or restarted) the server. However, in a production environment, it may be inconvenient to stop the running server with existing applications – that’s where Tomcat’s Manager web applications come in handy. Using Manager web application bundled with the Apache Tomcat distribution, you can deploy new applications, undeploy existing ones, and perform various other server management tasks from your browser.

In this chapter, we will:

- Define the Tomcat Manager web application, and discuss its usage
- Describe how to access the Tomcat Manager web application using its HTML web interface
- Explain the steps involved in accessing Manager’s text-based interface using Ant scripts

What Is the Manager Web Application?

The Tomcat Manager web application is packaged with the Tomcat server. It is installed in the context path of /manager and provides the basic functionality to manage Web applications running in the Tomcat server. Some of the provided functionality includes the ability to install, start, stop, remove, and report web applications. Using Manager web application, you can easily deploy web applications on the local or remote server, without the need for FTP access to the server itself – all commands are invoked over HTTP protocol – using your favorite browser or the command line.

Tomcat Manager web application shipped with Apache Tomcat 7 and has a web-based interface, so it can be accessed via any browser and commands issues by simply clicking on the links in the web page. Although a web-based browser is convenient and easy for human access, sometimes we need to manage web applications using external scripting tools. For that reason, commands to the Manager web application can be issued using the text-based interface.

For advanced access when security is critical, Manager web application can be accessed via Java Management Extension (JMX) proxy. JMX is a standard Java technology created for the management of applications, devices, or networks using standardized Java API. Because JMX is beyond the scope of this book, we won’t be getting into the details of JMX access to Tomcat Manager web application. For more information about JMX, see the JMX home page at
You can find details about JMX access to the Tomcat Manager application on Tomcat project’s web site at http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Using_the_JMX_Proxy_Servlet.

Finally, accessing status information of the Tomcat instance is also part of the Manager web application. You can see basic server stats from Tomcat’s status page, including the Tomcat version, Java version, OS version, and basic JVM memory and thread information.

For convenience and easier access and security management, these four ways of accessing Tomcat’s Manager web application are accessible on separate contexts (all sub-contexts of the main /manager context). Table 4-1 shows the contexts for accessing different parts of Manager web application.

<table>
<thead>
<tr>
<th>Component</th>
<th>Context Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML-based web app.</td>
<td>/manager/html</td>
</tr>
<tr>
<td>Text-based script</td>
<td>/manager/text</td>
</tr>
<tr>
<td>JMX proxy access</td>
<td>/manager/jmxproxy</td>
</tr>
<tr>
<td>Status request</td>
<td>/manager/status</td>
</tr>
</tbody>
</table>

In the following sections we will demonstrate issuing commands to the Manager web application using its HTML web-based interface, as well as using Ant script to access Manager web application from the command line.

**Gaining Access to the Manager Web Application**

Before you can use the Manager, you must set up a new user with the appropriate privileges to access the Manager web application. If you try to access any part of the Manager web application without setting up security privileges, you will be presented with a Tomcat error page, with standard HTML status code 403 - forbidden, as Figure 4-1 illustrates.