The creation of web applications using tomcat.

Required components

The Eclipse V3.2 Callisto integrated development environment (IDE) includes tools for doing Web development and integrating with servers. So, aside from the software development kit (SDK), you need only Eclipse and Apache Tomcat installed. The versions featured in this article are listed below.

Table 1. Components and versions used

Component Version used in this article

Eclipse Platform 3.2.2

Java™ EE 5 SDK 1.5.0\_09

Apache Tomcat 5.5.20

The Eclipse IDE is an open source IDE used as the development environment for your JavaServer Pages (JSPs) and Java files. With the Web and Java 2 Platform, Enterprise Edition (J2EE) Development tools installed, Eclipse is great tool for creating HTML, JSPs, and servlets.

You need the Java SDK to run servlets on your machine. Finally, Apache Tomcat is an open source Web and servlet container, used in the official reference implementations for Java Servlet and Java Server Pages.

Installation

Installation of Eclipse and Tomcat is similar — both are extracted from the archive file format into a folder on your hard disk. If you ever want to uninstall either, it's as simple as deleting the directory in which you extracted the files.

Install Eclipse

To install the Eclipse IDE, unpack the compressed (.zip or .tar.gz) file you downloaded from Eclipse, then move the unpacked folder into a convenient location. I try to install it in a location that follows the conventions of whatever operating system I'm using. If I'm installing Eclipse on a Macintosh, I put the Eclipse folder in the Applications folder. If I'm using Microsoft® Windows®, I put the unpacked folder into the C:\Program Files directory.

Once you've downloaded and installed Eclipse, start it up. Before you can create a Web project, you need to install the J2EE and Web Development plug-ins. Fortunately, it's easy to install the plug-in using the Callisto Discovery Site.

Install the Web tools

With the Eclipse IDE running, select Help > Software Updates > Find and Install. This option lets you download and install the Web tools without going to a Web site.

Select Search for New Features to Install, then click Next.

Select Callisto Discovery Site, then click Next.

Under Callisto Discovery Site, check the Web and J2EE Development box. The wizard immediately warns you that you're missing dependencies. Click Select Required, and these warnings should disappear. Some or all of the components in the Graphical Editors and Frameworks and Models and Model Development categories are selected because they're required for the installation of the Web and J2EE plug-ins.

Click Next to continue, and license agreements appear for each feature you've selected. If you agree with the licenses, accept them and follow the wizard through the rest of the process. When the IDE is finished installing the plug-ins, it asks you to restart Eclipse in order for the changes to take effect. Be sure you restart the IDE before continuing.

After you restart the IDE, the new features are available to begin developing dynamic Web projects.

Install Tomcat

To install Apache Tomcat, extract the files from the downloaded archive and place them into a directory. I put them in my C:\apps directory to make them easy to locate later. That's it for now; use Eclipse as shown later to start Tomcat.

Create a new Web project

From the Eclipse IDE, select File > New > Project to view the project wizards. Type Web in the Wizards box, and the New Project window filters the wizards to show the ones that match. This is an easy way to locate the wizard you want instead of going through each item.

Choose Dynamic Web Project from the list. You use the Standard Web Project Wizard to make static HTML Web projects. You can technically use Tomcat to run static Web sites, although a Web server such as Apache Web Server may be a better choice for static content. Web servers built specifically for serving up static pages tend to have less overhead and are tuned for that purpose.

Target a runtime

Under Target Runtime, you see <None>, because you haven't created a runtime yet for Apache Tomcat. Click New to open the New Target Runtime Wizard. Select Apache Tomcat V5.5 from the Apache folder, then click Next.

Now you have the opportunity to name your runtime, although the default name Apache Tomcat V5.5 is fine. Click Browse to locate the base folder for your Apache Tomcat configuration (mine is C:\apps\apache-tomcat-5.5.20). I recommend leaving the IBM® Java Runtime Environment (JRE) version the same as the workbench default JRE, unless you have a specific reason to change it. Once you've entered a valid directory, click Finish to create the runtime configuration.

You're asked if you want to switch to the J2EE Perspective. I always answer yes because that perspective includes views like the Servers view, which help to start and stop Tomcat later. You can always open the Servers view by selecting Window > Show View.

Add the server

In the Servers view is a view that contains nothing. This view is for application and Web servers; it lets you control them and monitor their status. To control Tomcat directly from the comfort of the IDE, add a reference to Tomcat by selecting New > Server from the context-sensitive menu. The New Server wizard, will default to the Apache Tomcat V5.5 runtime you just created.

Click Next. You're asked whether you want to add projects to the server. Select the project you created in the previous steps, then click Add.

Click Finish to close the wizard. Now the Servers view contains the Tomcat V5.5 Server with the project under it.

Create a sample page

The easiest way to test your new dynamic Web project and Tomcat server integration is to create a simple JSP and deploy it to Tomcat. You can create the new JSP file by selecting File > New > Other, or you can use the context-sensitive menu by right-clicking the project name in the Project Explorer view and selecting New > JSP.

Make sure the parent folder is WebContent and name the file index.jsp. Click Finish to create the page using the default template. If you click Next, you can choose different templates to use (such as HTML or XHTML) for the new page. The defaults are fine for this test.

Deploy the page

Now that the test page is complete, use Eclipse to redeploy it to Apache Tomcat. In the Servers view, notice that State is set to Republish. That is because changes have been made since the project was last deployed to the server. Right-click the server and select Publish from the menu. The state changes to Synchronized once the project has been deployed to the server.

http://www.ibm.com/developerworks/library/os-eclipse-tomcat/