REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Implementing and Publishing RESTful Web Services

1.1 The Java Options

Java offers more than way to implement and then to publish RESTful web services. This chapter explores some options. On the publishing side, the choices range from very basic, development-oriented tools such as the Grizzly RESTful container and the core `Endpoint` class; through lightweight, Java-centric web servers such as Tomcat and Jetty; and up to full-blown Java application servers (JAS) such as Glassfish, JBoss, and WebSphere. There are also various APIs for implementing RESTful services, both standard and third-party. Here is a short list:

- The `HttpServlet` and JSP APIs, introduced briefly in Chapter 1 and examined more thoroughly in this chapter.
- The JAX-RS (Java API for XML-Restful Services) API.
- The JAX-WS (Java API for XML-Web Services) API, in particular the `WebServiceProvider` interface.
- The third-party *restlet* API.

For the most part, the API does not constrain how the service is to published. The exception is the servlet API, as servlets need to be deployed in a servlet container such as Tomcat's Catalina or Jetty. (Jetty is the name of both the web server and its servlet container.) This chapter uses Tomcat and Jetty to publish servlet-based services. There are shotcuts for publishing JAX-RS and JAX-WS services but these, too, can be published with Tomcat or Jetty. Services based on the restlet API are meant to be published through a servlet container. The decision of how to publish depends on many factors, of course. For example, if service deployment requires wire-level security in the form of HTTPS together with user authentication/authorization, then a web server such as Tomcat is the obvious starting point. If the published web services are to interact with EJBs, which are deployed in an EJB container, then a souped-up web server such as TomEE (Tomcat with EE support) or a full JAS is the obvious choice. In development, simpler options such as Grizzly or `Endpoint` are attractive. This chapter introduces various options for publication; and Chapter 6 covers web services deployed in a JAS.