1.3.1 Java web概述

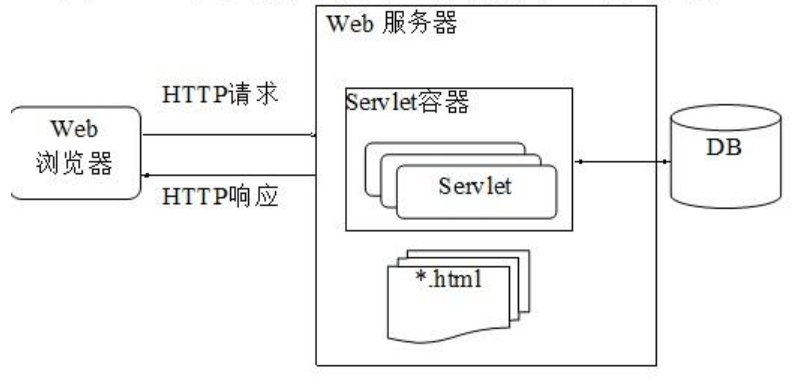
Java Web是用Java技术来解决互联网Web相关领域的技术总和，包括Web服务器和Web客户端两部分。Java在服务器端的应用非常丰富，比如Servlet、JSP和第三方框架等。Java技术对Web领域的发展注入了强大的动力。

1.3.2 Servlet 概述

Servlet可翻译成服务器端小程序，它是使用ServletAPI以及相关的类编写的Java程序，这种程序运行在Web容器中，主要用来扩展Web服务器的功能。Servlet自从1997年出现以后，由于所具有的平台无关性、可扩展性以及能够提供比CGI脚本程序更优越的性能等特征，使它的应用得到了快速的增长，并成为JavaEE平台的关键组件。Servlet技术实际上是CGI技术的一种替代。

1.3.3 Web容器

Web服务器使用一个单独的模块装载和运行Servlet与JSP页面，这个模块称为Servlet容器，或称Web容器。Tomcat就是一个Web容器，它在整个Web应用系统中处于中间层的地位，如图1-1所示



**图1-1 Web组件示意图**

图中给出Web应用系统的各种不同的组件构成，其中HTML文件存储在文件系统中，Servlet和JSP运行在Web容器中，业务数据存储在数据库中。

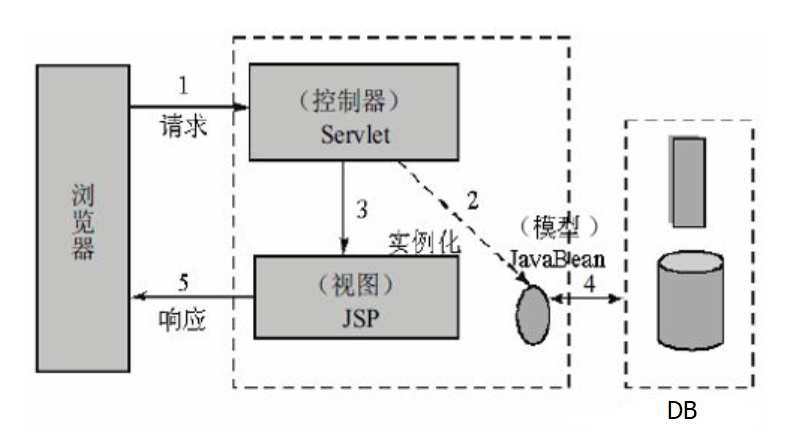
浏览器向Web服务器发送请求。如果请求的目标使HTML文件，Web服务器可以直接处理。如果请求的是Servlet或JSP页面，Web服务器将请求转发给Web容器，容器将查找并执行该Servlet或JSP页面，Servlet和JSP页面都可以产生动态输出。Tomcat作为一个Web容器可以有三种运行方式：独立运行的、进程内运行的和进程外运行的。独立运行方式是指Tomcat可以既作为Web服务器又作为Web容器运行。在这种情况下，整个Tomcat服务器包含两个模块：主Web服务器和Web容器。主Web服务器用来处理对静态内容的请求，Web容器用来处理对Servlet和JSP页面的请求。

1.3.4 JSP页面

JSP页面是在HTML页面中嵌入JSP元素的页面，这些元素成为JSP标签。JSP元素具有严格定义的语法并包含完成各种任务的语法元素，比如声明变量和方法、JSP表达式、指令和动作等。因此，JSP页面是一个由主动的JSP标签和被动的HTML标签混合而成的Web页面。在运行时，Web容器将JSP页面转换成Servlet类执行后将结果发送给客户。与其他的Web页面一样，JSP页面也有一个唯一的URL，客户可以通过它访问该页面。

1.3.5 MVC设计模式

在MVC设计模式的结构中，将Web组件分为模型、视图和控制器，每种组件完成各自的任务。在这种结构中所有请求的目标都是Servlet或Filter，它充当应用程序中的控制器。Servlet分析请求并将相应所需要的数据收集到JavaBeans对象或POJO对象中，该对象作为应用程序的模型。最后，Servlet控制器将请求转发到JSP页面。这些页面使用存储在JavaBeans中的数据产生相应。JSP页面构成了应用程序的视图。



**图1-2 MVC设计模式结构**

该模型的最大优点是将业务逻辑和数据访问从表示层里分离出来。控制器提供了应用程序的单一入口点，它提供了较清晰的实现安全性和状态管理的方法，并且这些组件可以根据需要实现重用。然后根据客户的请求，控制器将请求转发给合适的表示组件，由该组件来响应客户。这使得Web页面开发人员可一直关注数据的表示，因为JSP页面不需要任何复杂的业务逻辑。