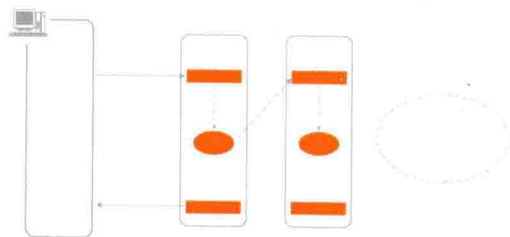
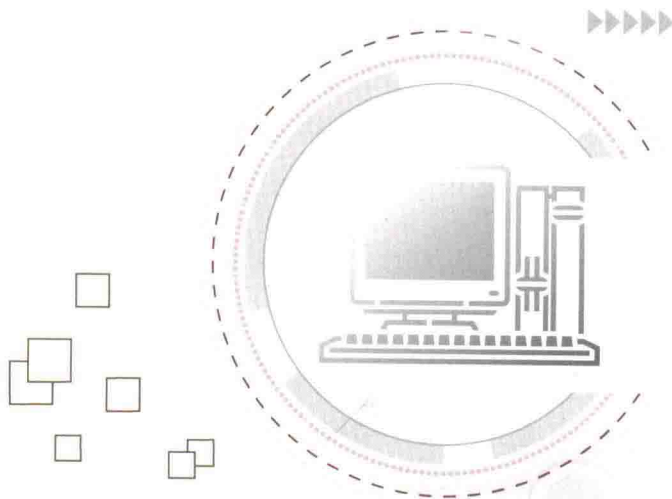




普通高等教育创新型人才培养规划教材

```
Connection con = null;  
Statement stm = null;  
ResultSet rs=null;  
try {  
    ...  
} catch (Exception e) {  
    ...  
} finally {  
    if(rs!=null){ rs.close();}  
    if (stm != null) { stm.close()  
    if (con != null) { con.close()  
}
```



# Java Web 编程技术

JAVA WEB BIANCHENG JISHU

刘启文 编著



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS



普通高等教育创新型人才培养规划教材

# Java Web 编程技术

刘启文 编著

北京航空航天大学出版社

## 内 容 简 介

《Java Web 编程技术》作为 Java Web 课程的教材,系统、全面地介绍了有关 Java Web 开发所涉及各类知识。全书共分 6 章,内容包括 Web 基础知识、Web 开发的客户端技术(包括 HTML、CSS 和 JavaScript 语言)、JSP 技术(包括 JSP 基础知识、脚本元素、指令元素、动作元素和隐含对象)、JDBC 技术(主要包括 JDBC 基础知识、JDBC API、事务处理、分页处理、批处理、大对象处理和 DAO 模式)、MVC 模式(主要包括 JavaBean、自定义标签、EL 和 JSTL)、高级技术(包括 Servlet 过滤器、Servlet 监听器和 Ajax 技术)。书中所有知识都结合具体实例进行介绍。

本书适合作为计算机科学与技术专业、软件工程专业及相关专业的本科教材,也可作为 Java Web 编程技术的培训教材,还可供从事 Java Web 应用开发的技术人员学习参考。

### 图书在版编目(CIP)数据

Java Web 编程技术 / 刘启文编著. -- 北京 : 北京航空航天大学出版社, 2016. 7

ISBN 978 - 7 - 5124 - 2145 - 5

I. ①J… II. ①刘… III. JAVA 语言—程序设计  
IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 120776 号

版权所有,侵权必究。

### Java Web 编程技术

刘启文 编著

责任编辑 梅栾芳

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:goodtextbook@126.com 邮购电话:(010)82316936

北京泽宇印刷有限公司印装 各地书店经销

\*

开本:710×1 000 1/16 印张:19.25 字数:410 千字

2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷 印数:2 000 册

ISBN 978 - 7 - 5124 - 2145 - 5 定价:42.00 元

---

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

# 前 言

Java Web 编程技术包含的知识点繁多,让许多初学者无所适从,不知道如何规划自己的学习路线。一般来说,掌握 Java Web 编程技术需要学习以下知识。

第一部分:Java 基础,包括 Java 语法、常见 API、集合框架、面向对象的概念等;数据库基础,尤其是 SQL 语言。

第二部分:Web 开发客户端技术,包括 HTML、CSS、JavaScript、DOM、Ajax 等;Java Web 开发技术,包括 Servlet、JSP、JDBC 等。

第三部分:常用框架技术,包括 Struts2、Spring 和 Hibernate;设计模式,包括 MVC 模式、DAO 模式、工厂模式等。

本书内容包含以上所说第二部分和第三部分知识,其中重点介绍第二部分知识,以及如何应用第三部分所介绍的设计模式。本书的内容承上启下,既可以引导初学者进入 Java Web 开发领域,又可以为学习各种框架技术和设计模式的开发者打下坚实基础。

全书分为 6 章:第 1 章介绍了 Web 基础知识,重点介绍 Web 客户端技术与服务器端技术的区别,以及 Tomcat 的安装与管理;第 2 章介绍了 Web 开发客户端技术,包括 HTML、CSS 和 JavaScript;第 3 章介绍了 JSP 技术相关知识,包括 Servlet 生命周期、JSP 生命周期、脚本元素、指令元素、动作元素和隐含对象;第 4 章介绍了 JDBC 技术,包括 JDBC 驱动程序、JDBC API、JDBC 基本开发流程、数据库连接池、事务处理、分页处理、批处理、大对象处理和 DAO 模式;第 5 章介绍了 MVC 模式,包括 JavaBean、自定义标签、EL、JSTL 以及基于 Servlet 的 MVC 模式开发流程;第 6 章介绍高级技术,包括 Servlet 过滤器、监听器和 Ajax 技术等。

本书从实战出发,采用案例教学,对所讲授的重要知识点,采用例子代码进行演示,图文并茂、一步一步地进行讲解,使读者能够快速掌握开发 Web 应用的方法。本书对 Java Web 技术的讲解不“浮于表面”,而是通过分析 Tomcat 自动生成的代码、Tomcat 的部分源代码讲解底层实现原理,例如 JSP 生命周期、JSP 语法、中文编码等。

本书适合作为计算机科学与技术专业、软件工程专业及相关专业的本科教材,也可作为 Java Web 编程技术的培训教材,还可供从事 Java



Web 应用开发的技术人员学习参考。

第一次写书,感觉好像是孕育生命,过程是漫长的、痛苦的,但是当书稿终于大功告成的时候,又感觉一切付出都是值得的。在写作的过程中,要感谢很多人的支持,首先感谢计算机学院的领导,没有领导的支持,这本书的出版可能还要延后;其次要感谢北京航空航天大学出版社的编辑,没有编辑的辛勤劳动,这本书也无法这么快顺利出版;最后要感谢我的家人,尤其是我的妻子,为我做好后勤保障工作。

特别要感谢在写作过程中对样稿提出建议和错误纠正意见的同事和朋友——安云哲、李晓明和赵连喜,以及对书中例子进行调试的各位同学——王硕、梅林、孙洪宇和王新宇。

由于作者水平有限,书中难免存在不足之处,敬请读者批评指正。

作 者

2015 年 12 月

# 目 录

第 1 章 Web 基础知识 .....	1
1.1 Web 的核心技术 .....	1
1.1.1 如何显示文档内容 .....	1
1.1.2 如何传递文档内容 .....	2
1.1.3 如何定位文档内容 .....	3
1.2 应用程序的开发模型 .....	5
1.2.1 单层开发模型 .....	5
1.2.2 两层开发模型 .....	5
1.2.3 三层开发模型 .....	6
1.2.4 N 层开发模型 .....	6
1.3 Web 的开发技术 .....	6
1.3.1 Web 的客户端技术 .....	7
1.3.2 Web 的服务器端技术 .....	7
1.4 Tomcat 安装与管理 .....	10
1.4.1 Tomcat 的安装 .....	10
1.4.2 Tomcat 的目录 .....	11
1.4.3 Web 应用程序目录结构 .....	12
1.4.4 Tomcat 的管理 .....	13
1.5 习 题 .....	15
第 2 章 客户端技术 .....	16
2.1 HTML .....	16
2.1.1 HTML 基础知识 .....	16
2.1.2 标题标签 .....	18
2.1.3 文本格式化 .....	18
2.1.4 超链接 .....	20
2.1.5 图像标签 .....	20
2.1.6 表格标签 .....	21
2.1.7 表单标签 .....	23
2.1.8 框 架 .....	26



2.2	CSS	26
2.2.1	CSS 基础知识	27
2.2.2	CSS 选择器	29
2.2.3	CSS 样式	32
2.2.4	CSS 盒模型	35
2.2.5	CSS 定位与浮动	38
2.3	JavaScript 语言	43
2.3.1	JavaScript 基础知识	43
2.3.2	基本语法	46
2.3.3	对象	53
2.3.4	DOM	73
2.3.5	BOM	79
2.4	习题	88
<b>第3章 JSP 技术</b>		<b>90</b>
3.1	JSP 基础知识	90
3.1.1	什么是 Servlet	90
3.1.2	JSP 的执行过程	97
3.1.3	为什么需要 JSP	102
3.2	脚本元素	102
3.2.1	表达式	103
3.2.2	声明	104
3.2.3	代码片段	107
3.2.4	注释	108
3.3	指令元素	110
3.3.1	page 指令	110
3.3.2	include 指令	113
3.3.3	taglib 指令	116
3.4	动作元素	116
3.4.1	forward 动作	117
3.4.2	include 动作	120
3.5	隐含对象	123
3.5.1	request 对象	125
3.5.2	response 对象	128
3.5.3	out 对象	134
3.5.4	session 对象	137



3.5.5 application 对象 .....	142
3.5.6 pageContext 对象 .....	145
3.5.7 page 和 config 对象 .....	149
3.5.8 exception 对象 .....	151
3.6 习 题 .....	159
<b>第 4 章 JDBC 技术 .....</b>	<b>160</b>
4.1 JDBC 基础知识 .....	160
4.1.1 JDBC 驱动程序 .....	161
4.1.2 JDBC API .....	162
4.1.3 JDBC 基本开发过程 .....	167
4.1.4 预编译语句 .....	176
4.1.5 调用存储过程 .....	182
4.2 JDBC 高级知识 .....	184
4.2.1 数据源与连接池 .....	184
4.2.2 事务处理 .....	188
4.2.3 批量处理 .....	191
4.2.4 分页处理 .....	193
4.2.5 大对象处理 .....	196
4.3 DAO 模式 .....	203
4.4 习 题 .....	215
<b>第 5 章 MVC 模式 .....</b>	<b>216</b>
5.1 JavaBean .....	216
5.1.1 JavaBean 规范 .....	216
5.1.2 JSP 与 JavaBean .....	218
5.2 标签与 EL .....	232
5.2.1 自定义标签 .....	232
5.2.2 EL .....	239
5.2.3 JSTL .....	242
5.3 基于 Servlet 的 MVC 模式 .....	249
5.3.1 从 Model 1 到 Model 2 .....	250
5.3.2 Model 2 开发流程 .....	251
5.4 习 题 .....	260





<b>第 6 章 高级技术</b> .....	261
6.1 Servlet 过滤器 .....	261
6.1.1 过滤器原理 .....	261
6.1.2 过滤器核心对象 .....	262
6.1.3 过滤器的开发与配置 .....	263
6.1.4 中文编码 .....	267
6.2 Servlet 监听器 .....	274
6.2.1 ServletContext 监听器 .....	274
6.2.2 HttpSession 监听器 .....	277
6.2.3 HttpServletRequest 监听器 .....	284
6.2.4 配置监听器 .....	285
6.3 Ajax 技术 .....	286
6.4 习 题 .....	296
<b>参考文献</b> .....	298

# 第 1 章 Web 基础知识

万维网(World Wide Web, WWW)是因特网(Internet)上的一个分布式应用,也是 Internet 上最流行的应用,万维网常被称为 Web。Web 的出现,促进了 Internet 的发展和普及,使 Internet 可以被广大的普通群众使用。

1989 年,英国人蒂姆·伯纳斯·李(Tim Berners Lee)在欧洲核子研究组织(European Organization for Nuclear Research, CERN)工作期间提出了 Web 的应用架构。1990 年,他在自己的机器上实现了第一个网站,1991 年 8 月 6 日,这个世界上的第一个网站正式发布。网站内容包括什么是 Web、如何使用浏览器以及如何建立一个 Web 服务器等。现在通过地址 <http://info.cern.ch/> 还可以访问这个网站的快照。

1994 年,万维网联盟(World Wide Web Consortium, W3C)在麻省理工学院计算机科学实验室成立,建立者是蒂姆·伯纳斯·李。W3C(<http://www.w3.org/>)是 Web 技术领域最具权威和影响力的国际中立性技术标准机构,拥有来自全世界 40 个国家的 400 多个会员组织。1994 年成立后,至今已发布近百项有关万维网的标准,对万维网发展做出了杰出的贡献。

本章主要介绍 Web 的核心技术、开发模型、开发技术、Tomcat 的安装和管理等知识;重点要了解 Web 开发技术和相关原理,能够熟练安装和管理 Tomcat;难点是能够理解 Web 的客户端技术与服务器端技术的区别。

## 1.1 Web 的核心技术

Web 最初的目的是管理 CERN 中分散的文档数据,例如报告、实验数据、个人信息以及电子邮件地址等。为了创建这种管理分布文档数据的统一平台,必须解决三个问题:如何显示文档内容、如何传递文档内容以及如何定位文档内容。这三个问题的解决方案,也构成了 Web 的核心技术。

### 1.1.1 如何显示文档内容

采用超文本标记语言(HyperText Markup Language, HTML)来显示文档内容,HTML 是目前网络上应用最为广泛的语言,也是构成 HTML 文档的主要语言。HTML 文档是由 HTML 命令组成的描述性文本,HTML 命令可以说明文字、图形、动画、声音、表格、链接等。HTML 制作的文档,叫做网页或 HTML 页面。HTML 的核心概念有两个:超文本和标记语言。



## 1. 超文本

超文本是用超链接的方法,将各种不同空间的文字信息组织在一起的网状文本。Web 是一种超文本信息系统,它的一个主要概念就是超文本链接,通过 HTML 中的超级链接实现文档之间的跳转,把分布的文档链接在一起。它使得信息不再是线性的,而是可以非线性地存储、组织、管理和浏览信息。

## 2. 标记语言

HTML 文档包括标签和显示内容两部分,标签是一套由角括号包含的关键字,不同标签有不同的含义,标签由浏览器负责解释执行。HTML 文档不能直接执行,必须在浏览器中运行,因此①HTML 文档的显示效果由浏览器决定,不同的浏览器,显示效果可能有差异;②HTML 文档的运行环境是浏览器,与计算机操作系统无关,这消除了不同计算机之间信息交流的障碍。

### 1.1.2 如何传递文档内容

Web 是 Internet 上的分布式应用,每次交互都涉及客户端和服务端,文档从服务器传递到客户端需要通过网络协议。

Web 所采用的协议是超文本传输协议(HyperText Transfer Protocol,HTTP)。HTTP 协议采用了请求响应模型,HTTP 协议永远是客户端发起请求,然后服务器进行响应。因此使用 HTTP 协议,无法实现服务器主动将消息推送给客户端。

HTTP 的工作流程如图 1-1 所示,分为 4 步:①创建连接。一个 HTTP 客户端(通常是浏览器)与 Web 服务器的 HTTP 端口(默认为 80)建立一个 TCP 套接字连接。②发送 HTTP 请求。浏览器向 Web 服务器发送一个文本的请求报文,报文由请求行、请求头部、空行和请求数据 4 部分组成。③返回 HTTP 响应。Web 服务器解析请求,定位请求资源。Web 服务器读取资源内容,包装成 HTTP 响应返回浏览器。响应由状态行、响应头部、空行和响应数据 4 部分组成。④关闭连接。Web 服务器主动关闭 TCP 套接字,释放 TCP 连接。浏览器解析 HTTP 响应协议包,首先查看状态行是否是成功的状态代码。然后解析每一个响应头,根据响应头中文档类型以及文档长度,读取文档内容并在浏览器窗口中显示。大部分情况下,返回的文档

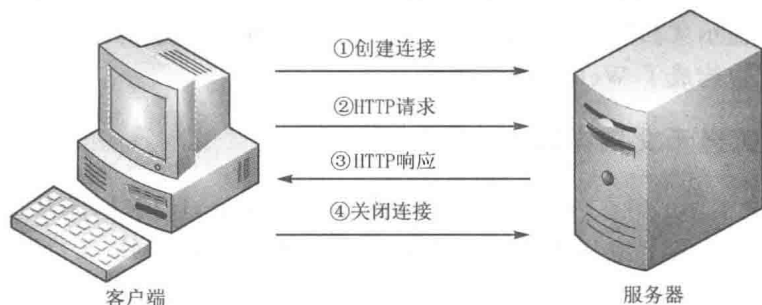


图 1-1 HTTP 工作流程



类型都是 HTML 文档。

Web 服务器需要面向全世界用户,因此可能需要同时处理百万、千万级客户端的网页访问,而且这种访问具有突发性和瞬时性,即两次访问请求的间隔会比较长,长时间保持网络连接,无法充分利用服务器资源,也无法适应大规模用户访问的特点。因此,HTTP 的设计者有意利用这种特点将协议设计为请求时创建连接、请求完释放连接,以尽快将资源释放出来服务其他客户端。HTTP 协议是一个无状态协议,即使是同一个客户端,每次发出的 HTTP 请求都是独立的,不同请求之间没有对应关系。

了解了 HTTP 协议的两个特点,即请求响应模型和无状态协议,以后做 Web 应用程序开发时,就知道哪些功能是能实现的,哪些功能是不能实现的。

### 1.1.3 如何定位文档内容

Web 所管理的文档在物理上是分布存储的,在逻辑上要求统一管理,所以这些文档需要统一命名,能够唯一识别。Web 中采用统一资源标识符(Uniform Resource Identifier, URI)对所有的文档进行标识。URI 有两种表示形式:统一资源定位符(Uniform Resource Locator, URL)、统一资源命名(Uniform Resource Name, URN)。URN 不依赖于位置,并且有可能减少失效连接的个数,但是需要更复杂软件的支持,无法在短期内广泛应用。目前广泛应用的是 URL,大部分情况下使用 URL 代替 URI。

通俗地说,URL 是 Internet 上用来描述信息资源的字符串,采用 URL 可以用一种统一的格式来描述各种信息资源。URL 的格式由三部分组成:协议、存有该资源的服务器地址(有时也包括端口号)和资源在服务器上的具体路径。第一部分和第二部分之间用“://”符号隔开,第二部分和第三部分之间用“/”符号隔开。第一部分和第二部分是不可缺少的,第三部分有时可以省略。

URL 的一般语法格式为(带方括号[]的为可选项):

```
protocol :// hostname[:port] / path / [;parameters][? query]# fragment
```

其中,①protocol 表示使用的传输协议,最常用的是目前 Web 中应用最广泛的协议——HTTP 协议。②hostname 表示存放服务器的域名或 IP 地址,有些服务器连接时需要进行用户身份验证,例如某些 ftp 服务器,这时可以在主机名前增加到该服务器所需的用户名和密码:username:password@hostname。③port 表示服务器进程绑定的端口,省略时使用协议的默认端口,如表 1-1 所列,大部分传输协议都有默认的端口号。如果某个服务器进程绑定到非默认端口,那么访问该服务器进程时,需要输入端口号。在同一台物理主机中,一个端口号只能绑定一个服务进程,当有两个以上服务进程绑定到同一个端口号时,会出现端口冲突错误。④path 表示资源在服务器上的路径,是由零或多个“/”符号隔开的字符串,一般用来表示服务器



上的一个目录或文件地址。⑤parameters 表示特殊参数,例如在 URL 中传递会话(Session)唯一编号,需要在这个部分设置。⑥query 表示传递参数列表,可有多个参数,用“&”符号隔开,每个参数的名和值用“=”符号隔开。⑦fragment 表示用于指定网络资源中片断的字符串。例如一个显示一篇长文章的网页,每个小节可以设置一个片断名,访问这个网页时,可以直接定位到某一个片断。

表 1-1 传输协议

名称	格式	默认端口
访问本地文件资源	file:///	无
访问 Web 服务器资源	http://	80
通过安全协议访问 Web 服务器资源	https://	80
访问 FTP 服务器资源	ftp://	21
访问支持 mms 协议的流媒体服务器资源	mms://	1755
访问支持 ed2k 协议的 P2P 服务器资源	ed2k://	4662
访问支持 thunder 协议的 P2P 服务器资源	thunder://	3076

下面结合一个例子进行分析,假设服务器 IP 地址 202.199.24.17,域名 www.sau.edu.cn,服务器上的 Web 服务进程绑定端口 80,默认访问的文件为 index.jsp。网站的部分目录结构如图 1-2 所示,网站的主目录为 D:\WebRoot,网站主目录又叫网站根目录,网站下的所有内容都存储在主目录或其中的子目录下,Images 即网站的子目录。如果要访问 index.jsp,则 URL 为 http://www.sau.edu.cn:80/index.jsp,因为 HTTP 的默认端口是 80,并且默认访问的文件是 index.jsp,因此访问 URL 也可以写成 ht-

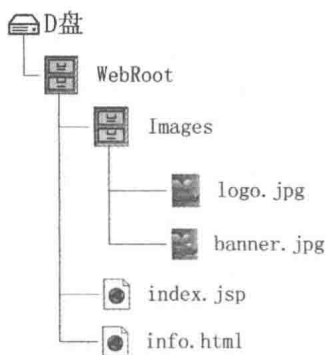


图 1-2 网站部分目录结构

tp://www.sau.edu.cn。如果要访问 logo.jpg,则 URL 为 http://www.sau.edu.cn/Images/logo.jpg。当然,可以使用 IP 地址替代域名。

下面是一个更复杂的 URL,要请求的资源是 index.jsp,本次 HTTP 请求所属会话的唯一编号为 5AC6268DD8D4D5D1FDF5D41E9F2FD960,同时向该资源传递的参数有两个,其中 w 参数值为 2,h 参数值为 3,则访问的 URL 如下:

```
http://www.sau.edu.cn/index.jsp;jsessionid = 5AC6268DD8D4D5D1FDF5D41E9F2FD960?
```

```
w = 2&h = 3
```



## 1.2 应用程序的开发模型

从逻辑上来说,大多数应用程序都由三部分组成:用户界面、应用逻辑和数据访问。用户界面是应用中直接面向用户的部分,主要完成数据的收集与展示,是应用程序中最贴近用户的部分,一般是与用户直接进行交互的界面。应用逻辑能实现应用业务流程处理,控制程序的执行路径,是应用程序中最复杂的部分。数据访问是应用中对数据进行管理的部分,主要完成应用程序对数据的增加、删除、修改、查询等工作。

随着应用需求的不断深化以及计算机技术的不断发展,应用程序的开发模型也在不断地发展和变化,先后出现了单层开发模型、两层开发模型、三层开发模型和N层开发模型等。前两种被称为传统的应用程序开发模型,后两种被称为 Web 开发模型。Web 开发模型成为当今应用软件的首选开发方式,对传统应用软件开发产生了深远的影响,出现了越来越多的基于 Web 的应用系统。

### 1.2.1 单层开发模型

最初的应用软件是单机软件,不管从物理上还是从逻辑上看,应用程序都运行在单个主机上。用户界面、应用逻辑和数据访问在一个应用中,功能紧密耦合在一起。因此①代码复用性、代码可维护性、代码的修改都十分困难;②应用程序不具有可伸缩性,可扩展性差;③应用程序所管理的数据无法共享。

这种单层开发模型在 20 世纪 70 年代及 80 年代初非常流行,但随着计算机和网络技术的发展,以及企业决策的分散化和信息资源的多元化,这种集中模式越来越难以适应现代化的需要。

### 1.2.2 两层开发模型

随着网络技术的发展,尤其是 Internet 的流行,在 20 世纪 80 年代中期出现了两层开发模型,即客户端/服务器(Client/Server, C/S)模式,应用程序从逻辑上分为两个部分(也许物理上还是在同一台机器上),处理代码分布在客户端和服务端之间。通过将任务合理分配到客户端和服务端,降低了系统的通信开销,可以充分利用两端硬件环境的优势。用户界面代码分配到客户端,数据访问代码分配到服务端,而应用逻辑代码可以根据不同情况分配到客户或服务端。如果应用逻辑代码只分配到客户端,则服务端只是一个数据库管理系统,不需要编写处理代码;如果应用逻辑只分配到服务端,那么需要编写一个专门的网络服务器,与客户端通信;如果应用逻辑分配到客户端和服务端,那么既可以编写一个网络服务器,也可以在数据库管理系统中通过存储过程的方式实现应用逻辑。

不管采用哪种方式,当应用需求发生变化时,都需要对客户端程序进行升级。当



客户端程序数量增多,部署范围增大时,维护成本会呈几何倍数增长。因此,虽然 C/S 模式因为其灵活性得到了极其广泛的应用,但对于大型软件系统而言,这种结构在系统的部署和扩展性方面还是存在着不足。

### 1.2.3 三层开发模型

在三层开发模型中,应用逻辑代码独立出来,分配到单独的业务服务器中,不再分配到各个客户端,降低了维护成本,提高了安全性。但是,因为客户端程序是定制开发的,当需要升级用户界面代码时,系统的维护成本还是随着客户端数量的增加而增加。

随着 Web 的发展,出现了浏览器/服务器(Browser/Server, B/S)模式。B/S 模式是一种特殊的 C/S 模式,即客户端固定为浏览器软件,服务器端固定为 Web 服务器。用户界面完全通过浏览器实现,Web 服务器中保存了用户界面、应用逻辑和数据访问代码,数据库服务器负责管理数据。

浏览器和 Web 服务器都是通用软件,不需要特别安装和维护。当用户访问 Web 应用程序时,通过浏览器向 Web 服务器发起 HTTP 请求,Web 服务器收到 HTTP 请求后,执行应用逻辑代码,然后返回执行结果,浏览器收到执行结果后进行显示。

所有的代码都保存在 Web 服务器上,因而系统维护非常容易,大大降低了系统的日常维护费用,并节约了开发成本。

### 1.2.4 N 层开发模型

从逻辑角度看,采用 N 层开发模型的系统,分成客户端、Web 服务器、应用服务器、数据库服务器四层;从物理角度看,应用服务器可以视用户并发数从 1 到 N 台进行扩充,以保证客户端用户的响应要求。

N 层开发模型中,每一层都可以被单独改变,而无需其他层的改变;降低了部署与维护的开销,提高了灵活性和可伸缩性;应用程序各部分之间松耦合,从而应用程序各部分的更新相互独立;业务逻辑集中放在服务器上由所有用户共享,使得系统的维护和更新变得简单,也更安全。

## 1.3 Web 的开发技术

Web 开发技术种类繁多,比较复杂。但是从 Web 开发模型来看,Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及到客户端和服务端两个层面,因此 Web 开发技术可以分为客户端技术和服务器端技术。Web 客户端是浏览器,所有在浏览器中运行的技术都是客户端技术;Web 服务器端是 Web 服务器,所有在 Web 服务器中运行的技术都是服务器端技术。但是单纯的 Web 服务



器只支持 HTTP 协议的解析,如果要运行复杂的应用逻辑代码,必须对 Web 服务器进行扩展,不同的厂商有不同的解决方案,因此构成了不同的服务器端技术。

### 1.3.1 Web 的客户端技术

Web 客户端,即浏览器端,主要执行用户界面代码,是用户与 Web 应用程序进行交互的接口。Web 客户端技术主要由三部分组成:结构、表现和行为。结构化标准语言主要包括 HTML、可扩展超文本标记语言(eXtensible HyperText Markup Language, XHTML)和可扩展标记语言(eXtensible Markup Language, XML),这些语言主要负责搭建 Web 页面结构;表现标准主要包括层叠样式表(Cascading Style Sheets, CSS)技术, CSS 主要负责控制 Web 页面的显示,使界面更美观;行为标准主要包括文档对象模型(Document Object Model, DOM)、JavaScript 等,这些技术主要负责在浏览器中执行一些应用逻辑。

Web 的客户端技术与所采用的服务器端技术无关,不管服务器端采用哪种开发技术,客户端技术都是一致的。Web 的客户端技术将在后面为大家详细讲解。

### 1.3.2 Web 的服务器端技术

Web 的服务器端技术有很多,主要是通过对标准的 Web 服务器进行扩展,增加应用逻辑的运行环境,不同的技术采用不同的语言实现,制定不同的访问接口。本节介绍公共网关接口(Common Gateway Interface, CGI)、超文本预处理器(Hypertext Preprocessor, PHP)、动态服务器页面(Active Server Page, ASP)、ASP. Net、Java 服务器页面(Java Server Pages, JSP)等技术,目前较流行的三种技术是 PHP、ASP. Net 和 JSP。

#### 1. CGI

CGI 是最早支持动态网页的服务器端技术,实现原理比较简单,应用逻辑代码由一个外部程序(CGI 程序)实现,然后由 Web 服务器与 CGI 程序进行数据交换。CGI 规定了 CGI 程序与 Web 服务器之间的接口标准,是在 CGI 程序和 Web 服务器之间传递信息的规范。CGI 规范允许 Web 服务器执行外部程序,并将它们的输出发送给 Web 浏览器。CGI 程序可以使用任何操作系统支持的编程语言进行编写,例如 Perl、Visual Basic、Delphi、C 或 C++ 语言等。

CGI 易于使用,但是有三个缺点:①编程效率比较低,因此 CGI 程序返回 HTML 文档,会在代码中混合大量的 Web 客户端技术(例如 HTML、CSS 和 JavaScript 代码等)。②维护升级复杂。如果采用需要编译执行的语言,例如 C 语言,那么每次修改程序后都必须重新将 CGI 程序编译成可执行文件。③Web 服务器根据用户发送的 HTTP 请求调用 CGI 程序,CGI 程序以进程的方式运行。因此,当多用户同时访问一个 CGI 程序时,会占用较多的服务器资源,造成服务器的响应速度下降。





## 2. PHP

PHP 早期其实是一个用 Perl 实现的 CGI 程序,于 1995 年由 Rasmus Lerdorf 创建,用来跟踪浏览其主页的访问者信息。经过多年不断的修改和版本升级,PHP 已经成为一种被广泛应用的、开放源代码的多用途脚本语言,可嵌入 HTML 文档中,尤其适合 Web 开发。PHP 可以完成任何 CGI 脚本可以完成的任务,与 CGI 相比,PHP 是将程序嵌入 HTML 文档中去执行,执行效率比完全生成 HTML 标记的 CGI 程序要高许多。

PHP 的优点:①语法简单,容易学习。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创的语法。②开放源代码,便于扩展,并且有成熟的开源社区提供的技术支持。③跨平台,便于移植,支持 Linux、Unix 以及 Windows 等操作系统。④执行效率高,性能卓越。Web 网站开发中占有率比较高,例如淘宝网、新浪、雅虎、163 等大型门户网站,都选用 PHP 来作为服务器端技术。

PHP 的缺点:①对多线程支持不好;②对于 N 层开发模型支持不好。

## 3. ASP

ASP 是微软公司开发的一个 Web 服务器端的脚本编写环境,利用 ASP 可以开发和执行动态的、互动的、高性能的 Web 应用程序。使用 ASP 可以组合 HTML 文档、脚本命令和 ActiveX 组件以创建交互的 Web 页和基于 Web 的功能强大的应用程序。ASP 程序运行环境是 Internet 信息服务(Internet Information Services, IIS)。ASP 既不是一种程序语言,也不是一种开发工具,而是一种技术框架。运用 ASP 可将 VBScript、JScript(JScript 是微软的 JavaScript 实现)等脚本语言嵌入 HTML 文档中,即可快速完成 Web 应用程序,无需编译,就可在服务器端直接执行。

ASP 的优点:①ASP 中使用的默认脚本语言 VBScript 直接来源于 Visual Basic,简单易学,容易上手;②开发和部署更加容易;③可以通过 ActiveX 组件实现更强大的功能,例如通过 ADO 组件可轻松访问数据库。

ASP 的缺点:①ASP 页面在服务器端是解释执行,运行效率不高;②ASP 只能在 Windows 系统中运行。

## 4. ASP.NET

ASP.NET 是 .NET 框架的一部分,是一项微软公司的技术,可用于在服务器上生成功能强大的 Web 应用程序,与其他服务器端开发技术相比,ASP.NET 具有以下几个优点:

### (1) 更高的执行性能

与传统 ASP 解释运行方式不同,ASP.NET 是编译后执行的,并且可利用早期绑定、实时编译、本机优化和盒外缓存服务等技术,使执行性能更高。

### (2) 更高的开发效率

Visual Studio 集成开发环境中增加了大量的工具箱和设计器,利用可视化编辑、拖放服务器控件和自动部署等技术,使 ASP.NET 的开发效率更高。