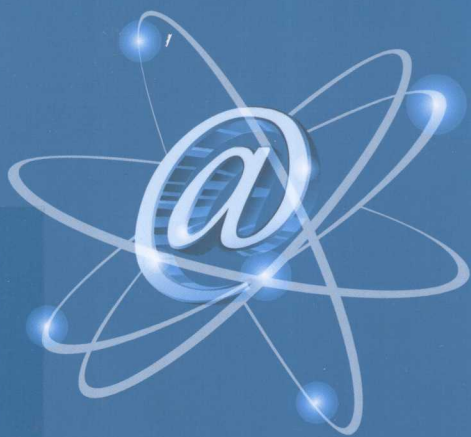


卓越工程师教育培养计算机类创新系列规划教材

Web 程序设计

主 编 王池社
副主编 马青霞 陆 悠



科学出版社

卓越工程师教育培养计算机类创新系列规划教材

Web 程序设计

主 编 王池社

副主编 马青霞 陆 悠

科学出版社

北 京

序

近年来，我国高等教育特别是计算机专业教育，正进行一系列与时俱进的大教学内容调整和教学方法改革，突出应用型人才培养目标，纷纷在计算机教育教学改革中强化实践教学，借助学以致用来培养学生实践能力和创新能力，不断地提升学生社会职业素养和就业竞争力。随着互联网的普及和应用加速，学习和掌握 Web 编程技术的期望越来越迫切，特别是对在校计算机专业大学生来说，Web 程序设计能力越来越成为现代计算机人才的一种必备能力。

针对当前高校计算机教育与就业岗位切合度不够，更多强调基础知识的传授，而忽视学生技能培养的诸多不足，教材编写团队结合多年从事 Web 编程教学所积累的经验 and 教学心得，并对市场上的 Web 编程教材进行调研，围绕如何加强学生实践能力的培养为编写突破口，科学设计教材内容和巧妙组织内容。

在教材中，充分考虑了 Web 编程技术人员的要求，将互联网编程中常用的技术融合到一起，让读者不仅能及时掌握 Web 程序设计基础知识，更为重要的是对实际项目编程中会用到的一些新技术进行了深入浅出的讲解，如 HTML 5、JavaScript、JS 框架、ASP.NET、基于 Web 的数据库访问技术、ASP.NET MVC 等，同时引入包括真实项目案例在内的许多实例，使得对技术的讲解不再枯燥，对学习 Web 编程的读者来说，具有重要的参考价值。

综上所述，编写实践强的教材很有意义也非常有必要，希望以后有更多的实际项目案例加入到这类实践教材中，编写出理论和实践有机结合的计算机类精品实践教材，培养出更多的优秀计算机编程人才。

南京信息工程大学教授、博士生导师



2016年4月20日

前 言

为了适应高校应用型人才培养迅速发展的趋势，服务社会，促进就业，着力提高学生的学习能力、实践能力和创新能力，出版了本书。

本书由拥有丰富教学和科研经验的教授编写，充分考虑了读者的知识背景。比较全面透彻地介绍包括 Internet 和万维网的起源与演变、客户端和服务端编程所需的工具和技术、建立和维护 Web 站点必需的工具和技术，讲述如何运用最新 Web 技术开发独立于平台的站点，包括 HTML 5、JavaScript、JS 框架、ASP.NET、基于 Web 的数据库访问技术、ASP.NET MVC 等。本书在内容上由浅入深，注重理论与实际的结合，综合案例精练，许多案例来自真实的科研项目，有利于培养学生综合分析问题和解决实际问题的能力，融知识性和实用性于一体。

本书的主要内容如下：

第 1 章主要介绍 Web 应用所涉及的基本概念、特点以及发展趋势，Web 应用程序的基本体系结构以及应用特性和 Web 应用开发所涉及的基本工具和平台。

第 2 章主要介绍 HTML 的概念、发展、应用、HTML 5 新特性和 HTML 5 新的表单处理等内容。

第 3 章主要介绍 JavaScript 的基本语法知识、常用核心对象、事件处理机制和 JSON 等相关技术。

第 4 章主要介绍 JS 框架在工程应用中的作用、主要类型，重点介绍 EasyUI 框架基本知识和使用。

第 5 章主要介绍 IIS 的安装与配置、创建 Web 应用程序、通过 IIS 发布 Web 应用程序以及 ASP.NET 常用内置对象的使用等内容。

第 6 章主要介绍数据库的基本概念，创建数据库及表的方法，数据库表的查询、增加、删除、修改的方法，掌握 .NET 数据库操作的基础知识以及掌握通过 ADO.NET 访问数据库等相关内容。

第 7 章主要介绍 ASP.NET MVC 的工作原理、项目的创建、项目中路由的配置、强类型视图的使用和 HtmlHelper 的使用等相关内容。

本书由王池社、马青霞、陆悠、张妮、吴宏杰、任勇、李尤丰等编写。

由于编者水平有限，书中难免存在不足之处，恳请广大读者批评指正。

编 者

2015 年 11 月

内 容 简 介

由于 Internet 技术的迅速发展, Web 应用和 Web 开发越来越普及, 已成为主流的应用开发方式之一, 本书结合教师多年的科研、教学和开发经验, 在内容上由浅入深, 注重理论与实际的结合, 综合案例精练, 许多案例来自真实的科研项目, 有利于培养学生综合分析问题和解决实际问题的能力。

本书共 7 章, 比较全面透彻地介绍包括 Internet 和万维网的起源与演变、客户端和服务端编程所需的工具和技术、建立和维护 Web 站点必需的工具和技术, 讲述如何运用最新 Web 技术开发独立于平台的站点, 包括 HTML 5、JavaScript、JS 框架、ASP.NET、基于 Web 的数据库访问技术、ASP.NET MVC 等。

本书可作为大中专院校学生学习 Web 程序设计和 Web 编程技术的教材, 也可作为广大自学者、软件开发人员和各类培训班的培训教材, 同时可供软件开发人员参考使用。

图书在版编目(CIP)数据

Web 程序设计 / 王池社主编. —北京: 科学出版社, 2016.8
卓越工程师教育培养计算机类创新系列规划教材
ISBN 978-7-03-047229-8

I. ①W… II. ①王… III. ①网页制作工具—程序设计—教材
IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2016) 第 012806 号

责任编辑: 邹 杰 / 责任校对: 桂伟利

责任印制: 霍 兵 / 封面设计: 迷底书装

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

文林印务有限公司 印刷

科学出版社发行 各地新华书店经销

*

2016 年 8 月第 一 版 开本: 787×1092 1/16

2016 年 8 月第一次印刷 印张: 17

字数: 403 000

定价: 43.00 元

(如有印装质量问题, 我社负责调换)

目 录

序 前言

第 1 章 Web 开发基础	1	2.3.10 应用程序缓存	41
1.1 Web 应用概述	1	2.3.11 Web Worker	43
1.1.1 Web 应用的起源	2	2.3.12 服务器发送事件	46
1.1.2 Web 应用的特点	3	2.4 表单新处理	48
1.1.3 Web 应用的发展历程与应用前景	4	2.4.1 新 Input 类型	48
1.2 Web 应用程序体系结构	5	2.4.2 新表单元素	52
1.2.1 Web 应用架构介绍	5	2.4.3 新表单属性	54
1.2.2 Web 应用的特性	8	2.5 小结	59
1.2.3 Web 应用前端内容开发技术：HTML 与 JavaScript 简介	10	第 3 章 JavaScript	60
1.2.4 Web 应用后端内容开发技术：Server 端技术简介	11	3.1 JavaScript 基础知识	60
1.3 Web 开发工具与平台	12	3.1.1 JavaScript 概述	60
1.3.1 前端开发工具	12	3.1.2 JavaScript 语法	62
1.3.2 后端开发工具	13	3.2 面向 Web 应用的 JavaScript 常用核心对象	83
1.3.3 应用服务器与数据库	13	3.2.1 浏览器对象	83
1.4 Web 应用开发工程师的知识体系结构	14	3.2.2 数组对象	87
第 2 章 HTML 5	16	3.2.3 日期对象	89
2.1 HTML 简介	16	3.2.4 数学运算对象	91
2.1.1 HTML 概念	16	3.2.5 字符串对象	93
2.1.2 HTML 的发展	17	3.3 面向 Web 应用的 JavaScript 事件处理机制	96
2.2 HTML 5 简介	17	3.3.1 事件机制原理	96
2.3 HTML 5 的新特性	17	3.3.2 event 对象	98
2.3.1 视频	17	3.3.3 事件处理机制实例——表单验证	102
2.3.2 Video + DOM	20	3.4 JavaScript 的文档对象模型	105
2.3.3 音频	21	3.4.1 文档对象模型的简介	105
2.3.4 拖放	23	3.4.2 获取文档的对象元素	107
2.3.5 画布	26	3.4.3 对 DOM 文档元素的访问	108
2.3.6 内联 SVG	31	3.4.4 JSON 技术	110
2.3.7 canvas 和 SVG 比较	32	3.5 综合应用实例	113
2.3.8 地理定位	32	3.5.1 基于 Web 的计算器应用	113
2.3.9 Web 存储	38	3.5.2 可视化日历控件应用	121
		3.6 小结	129
		第 4 章 JS 框架	130
		4.1 主要的 JS 框架简述	130
		4.2 EasyUI 框架	131

4.2.1	概述	131	6.3.6	触发器	211
4.2.2	EasyUI 框架在 VS 中的使用	131	6.4	ADO.NET 数据访问技术	213
4.2.3	EasyUI 主要组件介绍	133	6.4.1	ADO.NET 简介	213
4.3	综合应用实例——农民培训 监管系统	137	6.4.2	使用 Connection 对象连接 数据库	215
4.3.1	登录 UI 的实现	137	6.4.3	使用 Command 对象操作 数据库	221
4.3.2	主框架 UI 的实现	138	6.4.4	使用 DataSet 和 DataAdapter 对象操作数据库	224
4.4	小结	143	6.4.5	使用 DataReader 对象读取 数据	228
第 5 章	ASP.NET 基础	144	6.5	综合应用实例——成绩分析 系统	230
5.1	ASP.NET 概述	144	6.5.1	数据表	230
5.1.1	ASP.NET 简介	144	6.5.2	C#连接数据库	231
5.1.2	IIS 的安装与配置	144	6.5.3	系统的主界面	233
5.1.3	Visual Studio 2010 的基本操作	146	6.5.4	学生信息查询	235
5.2	ASP.NET 内置对象	150	6.5.5	教师管理成绩	236
5.2.1	Page 对象	151	6.5.6	教师分析成绩	238
5.2.2	Response 对象	152	6.6	小结	240
5.2.3	Request 对象	153	第 7 章	基于 .NET 框架的使用	241
5.2.4	Application 对象	156	7.1	ASP.NET 框架概述	241
5.2.5	Session 对象	157	7.1.1	ASP.NET Web Forms	241
5.2.6	Service 对象	158	7.1.2	ASP.NET MVC	242
5.2.7	Cookie 对象	160	7.1.3	两种框架的区别	243
5.3	小结	162	7.2	ASP.NET MVC 基本操作	243
第 6 章	数据库技术基础	163	7.2.1	创建默认的 ASP.NET MVC 的应用程序	243
6.1	数据库概述	163	7.2.2	创建自己的 ASP.NET MVC 应用程序	246
6.1.1	数据库简介	163	7.2.3	通过视图来呈现数据	247
6.1.2	数据库基本对象	166	7.2.4	使用 Model 来处理数据	249
6.1.3	常用数据库	168	7.3	强类型视图	251
6.2	数据库的搭建—— SQL Server 2012 简介	169	7.4	HtmlHelper 类	253
6.2.1	安装 SQL Server 2012	170	7.5	案例——ASP.NET MVC 实现 登录	255
6.2.2	SQL Server 2012 服务的启动 与停止	181	7.5.1	系统预览	255
6.2.3	SQL Server 2012 远程连接 的启用	182	7.5.2	数据库设计	256
6.3	SQL 入门	184	7.5.3	系统实现	256
6.3.1	数据库及表	185	7.6	小结	262
6.3.2	数据库的查询和视图	199	参考文献		263
6.3.3	数据库的备份和恢复	202			
6.3.4	SQL 常用函数	206			
6.3.5	存储过程	207			

第 1 章 Web 开发基础

本章知识要点

了解 Web 应用所涉及的基本概念、特点以及发展趋势；

掌握 Web 应用程序的基本体系结构以及应用特性；

了解 Web 应用开发所涉及的基本工具和平台；

了解 Web 应用开发所涉及的知识体系结构。

兴趣实践

了解 Web 应用开发的相关工具并进行安装，为后续章节学习提供基础。

探索思考

要学好 Web 应用开发这门课程，需要掌握哪些知识？根据知识体系结构思考个人的优势和不足在哪里，想一想这门课程应该如何进行学习才能确保质量和效率。

预习准备

请预先复习以前学过的 Windows 操作系统、SQL Server 数据库以及 C#语言的相关知识，特别是对 C#开发相关的内容需要着重进行预习。

1.1 Web 应用概述

进入 21 世纪以来，信息技术尤其是网络技术极大地改变了当今世界的生产和生活模式，从互联网、万维网到物联网、云计算，一个个新名词层出不穷，从简单的网页、多媒体浏览到电子商务、手机应用、社交软件，不断涌现的新业务逐渐地改变着人们的衣食住行，可以说，随着信息技术的飞速发展，人类社会进入了一个崭新的时代。

在这些对人类社会造成深刻影响的各种信息技术业务中，以互联网、Web 应用为代表的网络新业务占据着重要地位，尤其是 Web 应用，可以说，无论是从计算机终端到平板、手机的设备角度，还是从信息浏览、电子商务到游戏社交的应用角度，绝大多数的新兴业务都基于 Web 模式开发、部署并被人们使用。因此，如何深入学习和掌握 Web 应用开发的相关理论和技能，是摆在计算机专业学生以及爱好者面前的一项重要任务。

为了更好地展开对 Web 应用开发技能的学习，本节首先对与 Web 应用开发息息相关的两个基本概念——互联网与万维网进行简单说明。

1. 互联网

互联网(Internet)，又称网际网路，或被音译为因特网、英特网，是人们耳熟能详的一个名词，甚至人们常常将其视为现代信息技术及应用的代名词。但本质上互联网与网页浏览、电子商务等具体的 Web 应用并不相关，它仅仅是一个由分布于世界各个角落的网络互相串联

而构成的一个庞大网络，通过一系列标准的约定和规范(人们称其为网络协议)，这些网络互相之间能够实现数据信息的正确传输。而这样的一个由全世界几十亿个设备通过网络协议互相连接之后形成的逻辑上单一巨大的国际网络即互联网。互联网实质上是一个物理网络，在其基础上人们开发和部署了包括通信、数据传输等在内的众多应用系统，其中最广泛的一个信息系统即 Web 应用所对应的万维网。

2. 万维网

万维网(World Wide Web)，又称 Web、WWW、W3。与互联网不同，万维网并非是一个物理网络，而是一个由许多互相逻辑连接(而非物理连接)的特定格式的数据内容(人们称之为超文本，即 HTML 网页)组成的一个信息系统，该系统中的内容(包括网页、图像、多媒体数据等众多类型)被分布存储在全世界范围内的各个服务器中，而人们可以使用接入互联网的终端设备(计算机、平板设备、手机等)上的软件(人们称之为浏览器，英文名为 Browser)来访问这些内容。在这个信息系统中，每个超文本文件(包括图像、多媒体数据等内容)被视为一个“资源”，所有资源都由一个全域唯一的“统一资源标识符”(URL)来标志，而用户在浏览器软件中通过单击链接(也包括直接输入)等方式输入 URL 标识，然后浏览器使用超文本传输协议(Hypertext Transfer Protocol, HTTP)向相应的服务器发送请求，一旦服务器收到请求，它同样使用 HTTP 将资源返回给浏览器，最后浏览器读取返回的资源(HTML 文件)，并将其解析为可视化的形式展示给用户，此时用户就能在浏览器中看到丰富多彩的网页内容了。值得注意的是，在这个访问过程中，浏览器发送请求以及服务器传输资源时，数据是通过互联网进行传递的。

由此可见，互联网并不等同于万维网，互联网是万维网的物理基础，也是其承载者，而万维网则是部署于互联网上也许最为重要的一个应用系统，是互联网所能提供的服务其中之一。而人们日常使用的各类新闻浏览、电子商务等林林总总的应用则是当今不断发扬壮大的万维网中的一个有机组成部分。

思考：有人说互联网、万维网、因特网都是一回事，仅仅是功能上有差别，你是否同意？

1.1.1 Web 应用的起源

见惯现在美轮美奂的 Web 应用的人们很难想象，世界上第一个 Web 网站是多么的简陋，甚至都没能完整地保留下来，人们只能通过 <http://info.cern.ch> 这一网址大致浏览其内容。图 1-1 所示即世界上第一个 Web 网站的大致内容。其实世界上第一个 Web 网站的地址是 <http://info.cern.ch/hypertext/WWW/TheProject.html>，其后该地址几经更改，最终变为 <http://info.cern.ch> 并一直沿用至今，现在这个网站只是第一个 Web 网站在 1992 年版本的副本，并在原有基础上有所更改。

Web 技术的创始人是英国计算机科学家蒂姆·约翰·伯纳斯-李(Timothy John Berners-Lee)爵士，在 1955 年 6 月 8 日出生于英格兰伦敦西南部。他的父母参与了世界上第一台商业计算机(曼彻斯特 I 型，Manchester Mark I)的建造，因此在家庭的影响下，他对计算机技术有着非凡的热情和敏锐的洞察力。

1980 年 6~9 月，伯纳斯-李加入 CERN(欧洲核子研究组织)。由于 CERN 是一个规模庞大、雇佣了大量的科学研究人员一起协同工作的组织，伯纳斯-李敏锐地发现，如何让这么多科学家快速、方便地互相分享数据以及研究成果是 CERN 组织所面临的一个非常重要的问题。

面对这一挑战，伯纳斯-李提出了一个构想：创建一个以超文本系统为基础的信息系统，方便研究人员分享及更新信息。所谓超文本，实质上是一种用户接口技术，简单来说，就是在显示文本及与文本相关的内容时，允许其中的某些文字包含所谓的超文本链接，这种链接指向其他字段或者文档，在阅读软件的支持下，允许用户从当前阅读位置直接切换到超文本链接所指向的文字，从而增强信息浏览的方便性。同年，伯纳斯-李创建了一个原型系统 ENQUIRE 来验证他的构想，但这个系统并没有获得人们的重视，并且伯纳斯-李本人也在不久之后离开 CERN 到一家计算机公司(约翰·普尔图形计算机系统有限公司)任职，幸运的是，在该公司期间，他接触了计算机网络技术并得到大量的互联网方面的经验，这为他在互联网基础上进一步完善和深化自己的构想提供了基础。

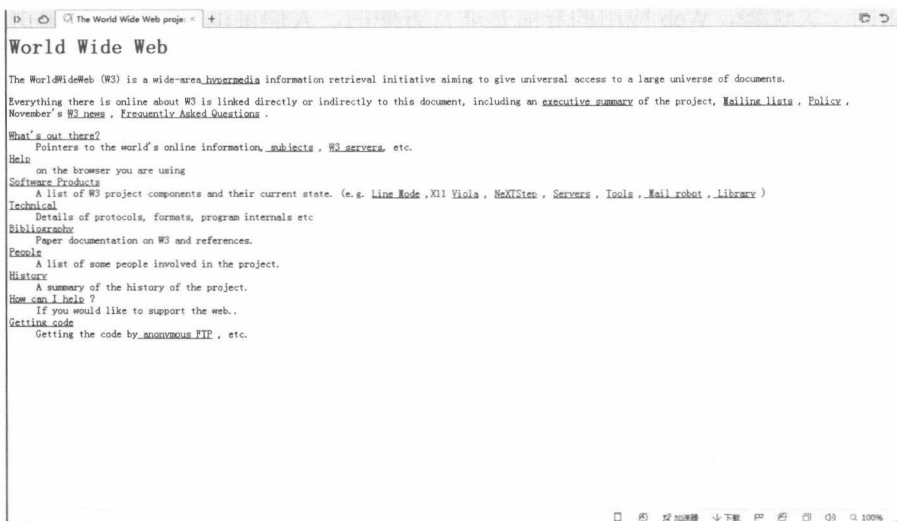


图 1-1 世界上第一个 Web 网站的大致内容

到了 1984 年，伯纳斯-李重返 CERN。此时互联网技术已经日趋成熟，而随着规模的进一步扩大，CERN 已然成为当时全欧洲最大的互联网节点。伯纳斯-李因此看到了将超文本系统与互联网结合在一起的机会。经过长时间的努力，1989 年 3 月，伯纳斯-李写下了他关于万维网的初步构想，在同事罗伯特·卡里奥的帮助下，他于 1990 年 11 月正式提出万维网的建议，并获得他当时主管的同意，然后在 1990 年的圣诞假期，伯纳斯-李设计了开发部署第一个 Web 站点所需要的技术工具：世界上第一个 Web 浏览器(同时也是网页编辑器)和第一个 Web 服务器。

最终，在 1991 年 8 月 6 日，世界上第一个 Web 站点正式上线。该网站部署于一台位于 CERN 的 NeXT 计算机。在该站点中，浏览者能够更多地了解超文本系统及技术细节，甚至能得知如何在网站上查找信息。另外就在同一天，伯纳斯-李还在 alt.hypertext 新闻组上发布了万维网项目简介的文字，这一天标志着万维网的首次亮相。从那一天起，万维网开始茁壮成长，不断发展，日趋深远地影响着整个世界。

1.1.2 Web 应用的特点

相对于其他信息技术，Web 应用有着自身的特点，掌握和了解这些特点能够更好地帮助我们学习和理解 Web 应用开发技术，下面简单介绍 Web 应用的几个特点。

1. 图形化界面

Web 应用之所以能够在世界范围内得到广泛应用和普及，其图形化显示的界面是首要原因，由于超文本能够将文字、图形甚至音频、视频有机地结合在一起，这极大地提高了人们在阅读信息时的体验，也使得 Web 应用能够承载比传统书籍、电影电视、唱片等媒介更为丰富的信息内容，使人更易阅读同时更乐于阅读。因此，如何合理地设计和实现 Web 的图形化内容是 Web 应用开发者首先需要掌握的技能。

2. 非线性的内容组织

Web 应用所使用的超文本除了能整合不同类型的信息内容，其非线性的内容组织也是吸引人们使用的一大优势，Web 应用的导航是非常方便的，人们可以轻松地通过单击链接从一处内容直接跳转至其感兴趣的另一处内容而无须做更多操作，与此同时，对内容提供者而言，这样的非线性组织方式也为其整合众多信息数据带来了方便。对于一个 Web 应用来说，如何为其内容设计合理的链接关系以保证良好的用户体验，也是摆在开发者面前的一项艰巨任务。

3. 访问者与平台无关性

对 Web 应用而言，用户的访问必须通过浏览器软件来实现，而相对于其他网络应用中的客户端往往非常复杂而言，浏览器的开发较为简单，并能够在包括 Windows、UNIX 甚至手机 Android 以及 iOS 等众多平台上实现，因此借助浏览器，用户能够实现与平台无关的 Web 访问，与此同时，Web 应用开发者也可以避免为不同平台开发不同应用这一弊端，开发成本大为降低。而在开发 Web 应用时，开发者也需在外观、内容等设计时充分考虑不同平台之间的兼容性，从而保证平台无关性。

4. 内容的分布式存储

对于应用提供者而言，由于超文本的特点，可以不用在本地耗费大量资源存储网页涉及的所有图形、音频、视频等信息资源，可以将这些资源分别存放在不同的站点上，而通过超链接将这些资源在逻辑上整合为一个整体提供给用户，这使得在世界范围内整合信息资源、节约开销成为可能。

5. 动态性与交互性

随着表单以及服务器端技术的不断发展，当前的 Web 应用已能够向用户提供足够的交互能力，用户可以提交自己的请求，服务器按需进行计算，最终动态生成内容返回用户。这种动态和交互的特性使得 Web 应用能够再现当今社会大部分生产生活流程，从而方便设计人员开发各种应用。而从社会实际生产生活需求出发，将其解析为相应的浏览器端以及服务器端的操作流程，并最终将其转化为代码的设计、实现以及部署是 Web 应用开发的主要内容，也是本书的主要内容。

1.1.3 Web 应用的发展历程与应用前景

虽然第一个 Web 网站非常简陋，但从它诞生的那一刻开始就引起了人们的重视，随着信息技术、互联网技术的飞速发展，人们对 Web 应用的需求越来越大，也越来越复杂，相应而来的是 Web 应用技术不断进化。根据应用特点，Web 技术的发展历史大致可以分为以下三个阶段。

1. Web 1.0 时代

其时间跨度最大, 大约从 Web 网站诞生直到 21 世纪初, 覆盖了整个 20 世纪 90 年代。其标志事件包括 1993 年推出的 HTML 1.0, 1995 年推出的 HTML 2.0, 1997 年推出的 HTML 3.2, 以及 1997 年推出的 HTML 4.0 等一系列 HTML 版本。在这个阶段, Web 应用是以内容为核心, 静止、单向、被动为其特点。Web 1.0 时代的网站大多使用静态网页 (HTML 4.0 及以下) 向用户提供各种内容 (新闻、文本、多媒体等), 而内容则是由网站开发者生成并向用户单向发送的, 用户处于静态的、被动的接收地位。用户所能进行的主动行为主要是搜索自己感兴趣的内容。因此这个阶段中, 以 Yahoo、搜狐乃至 Google、百度等为代表的搜索引擎成为贡献最大、最受欢迎的技术公司。值得注意的是, 虽然这个阶段以静态网页为主流, 但并不意味着整个万维网只有静态网页, 在该阶段已经出现了用来接收用户输入的表单等动态网页技术, 并得到了广泛应用, 只是这个阶段整个万维网中的内容仍以静态为主流。

2. Web 2.0 时代

时间跨度较为接近当今, 大约从 21 世纪初到目前, 其主要标志包括 2014 年通过的 HTML 5.0 草案。Web 2.0 阶段不再以 Web 网站而是以用户为内容核心, 人们可能已经发现, 当前网络中的内容更多的是以用户自己生成 (User-generated Content, 如微博、博客、用户上传的视频、图片等) 为主, Web 网站更多的是负责提供各类 Web 应用来接收、存储、组织用户生成的内容, 并提供各种渠道来方便用户之间的交流。与 Web 1.0 相比, Web 2.0 的各种服务更专注于用户, 提供用户之间更好的交互体验。在这个时代, 以 Facebook、腾讯、YouTube、优酷等注重社交网络、用户内容分享的技术公司贡献最大也最受人们欢迎。值得注意的是, 在这个阶段, 以手机、平板电脑为代表的移动终端已成为互联网中最为重要的组成部分, 因此 Web 应用的跨平台特性也得到了相应的发展, 通过手机访问能获得和台式机、笔记本终端相同的使用体验已成为现实。

3. Web 3.0 时代

这是目前正在进行的一个发展阶段, Web 3.0 一词目前还没有得到一个公认的定义, 人们提出了多种含义来阐述 Web 3.0, 分别用来概括可能出现的各种不同的方向和特征。总体来说, Web 3.0 的主要目标应该是让 Web 网络成为用户需求理解者和提供者, 通过引入人工智能、大数据处理以及语义网络等技术, 实现人和网络之间的顺利沟通, 进而进一步提高人与人沟通的便利性。在 Web 3.0 时代, 网络将能够对用户的需求了如指掌, 能够分析、掌握用户的行为习惯, 进而通过数据分析、资源筛选以及智能匹配等技术, 直接向用户提供其所需要的各种内容。总之目前来看, Web 3.0 还充满了争议和分歧, 在 Web 2.0 日益健全完善的今天, 什么才是 Web 3.0, 也许时间才能给我们答案。

思考: 请说出自己最常用的 3 个 Web 应用业务, 并比照上述标准, 想想这些业务属于哪一种。

1.2 Web 应用程序体系结构

1.2.1 Web 应用架构介绍

Web 的访问过程从本质上可以看成用户使用浏览器 (Browser) 访问服务器 (Server) 的过程, 因此 Web 应用的架构被称为 BS (Browser-Server) 架构, 其中浏览器是通用软件, 相对较

为简单，处理用户业务的复杂应用全部部署在服务器端，根据服务器端的不同以及 Web 应用的发展历程，Web 应用大致可以分为如下几种从简单到复杂的架构。

1. 客户端-网页服务器模式

这是最简单的访问架构，具体如图 1-2 所示，用户使用客户端(浏览器)向服务器发出请求(输入 URL，图 1-2 中的 URL 为 `http://www.sina.com.cn/index.html`，这段代码中 `http` 表示使用的协议为 HTTP；`www.sina.com.cn` 为服务器的域名，表示存储资源的主机位置；`index.html` 是资源名)，表示用户需要网站内的某个资源(网页、图片等)，该请求通过网络发送到网页服务器(由互联网的相关协议来保证这个请求最终会被送到 `www.sina.com.cn` 主机处)，服务器根据其请求的文档名称(`index.html`)，在自身文件系统中查找到文档并将其传输回客户浏览器，浏览器将文档以可视化形式展现给用户。在这个架构下，用户进行的是静态的网页浏览，此阶段的网站还不具备如今 Web 应用中普遍具有的如提交表单等复杂的与服务器端之间的交互能力。



图 1-2 客户端-网页服务器模式

2. 客户端-应用服务器模式

模式 1 下用户可以进行的操作仅仅是请求静态网页，然而随着 Web 应用的逐渐普及，很多网站希望能够给予用户一定的交互能力，使得网站能够更为灵活地向用户提供内容(例如，用户可以注册后登录，然后根据用户注册时提供的兴趣爱好，网站显示相关的内容，从而提高用户体验)，因此技术人员对网页服务器进行改进，使得能够在其上部署程序(通常使用通用网关接口(Common Gateway Interface, CGI)，以及超文本预处理(Hypertext Preprocessor, PHP)脚本等语言工具编写)处理用户动态传递的内容，进而有针对性地提供内容，与此相应，浏览器端用户也能够使用表单等工具向网站服务器所部署的程序传递参数，从而使得 Web 应用开始具备了服务器与客户端的交互能力。该架构如图 1-3 所示，图中用户在访问 `www.sina.com.cn` 主机时提交了登录的参数(用户名参数 `user`，值为 `tom`，用户密码参数 `pwd`，值为 `123`)，应用服务器在接收这两个参数后进行判断，若是合法用户，则根据用户信息动态生成相应的信息内容，这些信息会以 HTML 文件形式返回至用户的浏览器供用户查看。

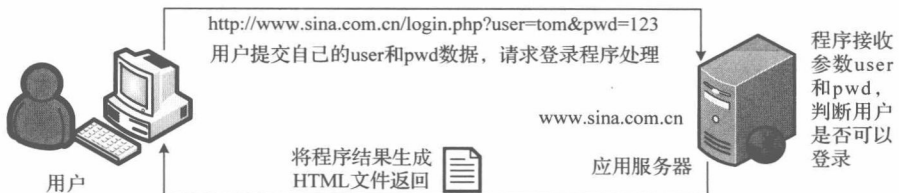


图 1-3 客户端-应用服务器模式

3. 客户端-网页服务器、应用服务器模式

基于模式 2 的 Web 应用能够处理用户提交的参数,但这种模式中应用服务器会对所有用户的每一次请求都进行判断,如果是动态内容的访问就进行相应的处理,然而一个 Web 应用并非所有内容都需要与用户交互,相对来说仍有大量静态文档会成为用户访问的目标,这样,这些静态页面的请求对应用服务器来说其实是无须处理的,因此为了提高效率,人们提出将提供静态内容的网页服务器和负责与用户交互的应用服务器分离,该架构如图 1-4 所示,Web 站点使用网页服务器接收用户所有的请求,当请求为静态页面时直接返回相应文档,当请求为动态内容时则转交第二层的应用服务器处理,应用服务器处理完毕后内容返回网页服务器继而返回用户。由于静态页面请求相对来说占的比例更大,因此这种架构能够有效提高对用户的响应速度,改善用户体验。而这种将不同目的和特性的服务器分离的思想在随后的 Web 应用发展过程中也得到了普遍应用。



图 1-4 客户端-网页服务器、应用服务器模式

4. 客户端-网页服务器、应用服务器、数据库模式

随着 Web 应用类型和规模的增长,很多复杂的类似信息管理系统(MIS)的信息处理业务逐渐被转移到 Web 应用中来,这些业务的一个突出特点是需要大量的数据库检索、更新等操作。对于这些 Web 应用来说,模式 2、模式 3 架构中的应用服务器在处理能力上有所不足,因此人们进一步应用服务器分离的思想,将数据库服务器引入,专门负责数据的存储、检索、更新等操作,从而增强应用处理信息检索方面的能力,该架构具体结构如图 1-5 所示,与模式 3 比较相似,负责接收用户请求的仍然是网页服务器,但处理动态内容请求的则是应用服务器和数据库。



图 1-5 客户端-网页服务器、应用服务器、数据库模式

5. 客户端-多层服务器模式

与前述架构主要针对业务本身日益复杂的趋势不同,客户端-多层服务器模式主要针对用

户规模逐渐扩大的趋势，随着 Web 应用的普及，许多网站的用户数量甚至呈指数级增长，面对如此规模的用户数量，单个服务器已不堪重负，因此人们采用增加同类服务器数量，构建集群一起向用户提供服务的思路，具体架构如图 1-6 所示，无论是应用服务器还是数据库服务器都呈层次架构，每个服务器负责一部分功能(或者一部分用户)，所有服务器最终构成一个整体向用户提供服务。

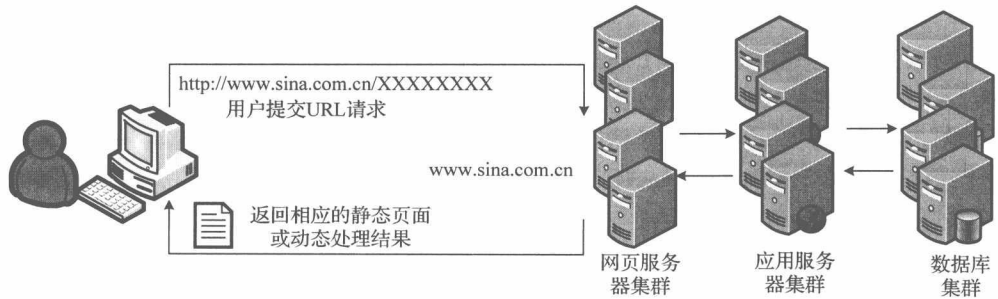


图 1-6 客户端-多层服务器模式

6. 客户端-Web 服务器、数据库模式

在实际应用中，对于大多数中小型规模的 Web 应用而言，同时使用网页服务器、应用服务器分别应对用户的静态、动态内容的请求成本过高，为了简化开发、部署方面的开销，人们在模式 3 的基础上，提出使用能够兼具网页服务器与应用服务器功能的 Web 服务器(较为常见的产品包括 IIS、Tomcat 等)，为了适应当前 Web 应用中对数据检索、更新的需求，同时采用数据库，从而构成图 1-7 所示的客户端-Web 服务器、数据库模式，本书的内容也主要是基于该模式展开的。



图 1-7 客户端-Web 服务器、数据库模式

思考：请分析下列 4 种 Web 业务更适合哪种 Web 应用架构来实现，并说明理由。

- (1) 向全校学生提供摄影作品浏览的网站。
- (2) 面向全校学生的图书馆管理系统。
- (3) 在全国范围内，以新闻浏览为主，提供用户实名评论的新闻网站。
- (4) 类似淘宝、亚马逊等电子商务网站。

1.2.2 Web 应用的特性

Web 应用之所以采取 B/S 架构，究其原因，主要是为了实现 Web 浏览客户端与 Web 服务器端之间的解耦合。其实在网络发展初期，很多应用并没有采用 B/S 架构，而是采用 CS (Client-Server) 架构，该架构的最大特点是客户端承担了较为复杂的业务功能，且与服务

器端功能紧密结合,对这些应用来说,用户必须使用特定的、功能较为丰富的客户端(人们称其为胖客户端,如 QQ 聊天的客户端、魔兽世界的游戏客户端等),客户端使用专用的网络应用协议与服务器端进行通信,传输用户的操作,调用服务器端的应用软件进行处理,然后客户端接收返回结果并通过人机接口(用户界面)告诉用户。这种模式下用户客户端可以实现比较复杂的功能,完成很多特定的应用(如网络游戏等需要用户复杂操作的应用)。但这种模式的一大弊端是客户端与服务器端结合得太紧密了,以至于服务器端的应用软件如果需要修改,往往要求用户的客户端也进行修改,而客户端分散在海量规模的用户处,因此造成客户端维护开销巨大(每次网络游戏升级版本时往往需要大量人力、物力去帮助玩家更新,一旦用户数量较多,更新体验往往较差)。

针对这种应用客户端与服务器端太过紧密的弊端,Web 应用采用了不同的思路,它采用了通用的客户端(浏览器)和通用的服务器端(各种 Web 服务器软件),其中通用的客户端功能较为简单(人们称其为瘦客户端),而复杂的业务只需部署在通用的服务器之上(无论是静态的文档页面还是动态的程序内容),用户操作通用的浏览器就一定能够正确地访问。这就使得服务器端的应用升级更新可以完全脱离客户端,从而使得 Web 应用能够实现比传统网络应用更快速的开发和迭代更新,从而更及时灵活地响应用户的需求。

这种 B/S 架构需要一系列技术的支撑,首先是用户浏览器需要实现人机接口界面,为了保证通用性,人们提出 HTML 来描述和构建网页内容,只要浏览器符合规范,能够正确地解读 HTML 文档,那么使用不同终端的人们看到的网页就是一致的;其次所有的网页内容存放在服务器上,每次用户需要使用 Web 应用时都需要通过网络请求服务器将网页发送至客户的浏览器端,这就保证了 Web 应用的内容更新时不需要用户浏览器参与,每次用户访问到的始终是最新版本的应用页面;最后则是规定了浏览器与服务器端之间的通信协议,即 HTTP,在该协议规定下,用户浏览器与服务器之间使用格式化的字符流互相交流,用户的各种操作,对应用程序提交的参数都会转化为相应的字符流,从而保证浏览器的通用性,简化服务器端应用开发的复杂度。

值得大家重视的是,这种 B/S 架构还有一个非常特别的特点,就是用户使用浏览器访问应用时存在“连接不保持”特性,简单说明如下。对于 Web 服务器来说,用户使用浏览器访问 Web 应用的过程可以看成是一系列的单击(Click,包括打开网页,单击页面中的某个超链接,再单击新页面中的某个超链接,以此类推,另外填写并提交表单实质上也可看成是单击某个超链接),而每次单击都会产生一个访问请求(URL 请求),这样,一个用户的操作实质上就会形成一系列请求,通过 HTTP 从浏览器出发依次到达 Web 服务器,但是对 Web 服务器来说,它不会把来自同一个用户的一系列请求作为一个整体来处理,而是把它们当成一个个单独的访问请求,每一个访问请求处理完之后立即发回结果。这么做的原因是:一方面因为 Web 服务器需要同时面对大量的用户,如果需要“记住”同一个用户的一系列操作,就需要为每个用户建立相应的存储结构等资源,这会导致服务器资源非常紧张,另一方面也是因为用户浏览网页的动作习惯并不是高速连续地请求多个网页,往往两个网页请求间有着较长的时间间隔,这就进一步增加了“记住”用户的开销。因此,服务器采用了这样的“连接不保持”技术来提高自己对大量用户的应对能力。

这样的服务模式在面对 Web 1.0 这种以静态页面访问为主的应用尚不存在什么问题,但对用户交互较为复杂的 Web 2.0 应用来说,这种服务模式就面临不足,例如,用户在第一次访问时已经提交了自己的用户名和密码,那么他必然希望后续访问中服务器能够“记住”自

己已经提交的信息而不用重复输入，然而服务器将该用户第一次输入的访问与后续的访问视为各自独立的访问，从而导致 Web 应用没法“记住”用户名和密码，为了解决这个矛盾，人们提出了 Session 以及 Cookie 等技术，这部分内容在 1.2.4 节中会具体介绍。

1.2.3 Web 应用前端内容开发技术：HTML 与 JavaScript 简介

根据 Web 应用所采取的 B/S 架构的特点，Web 应用的工程开发涉及的技术大致可以分为两大部分，其中，被加载到用户客户端（浏览器）、用户可以查看和操作的部分被称为前端内容，通常来说，前端内容相当于应用所具备的用户界面；而用来产生前端内容的业务逻辑和数据处理部分则是后端内容，这部分内容往往不被用户直接感知，但它们是 Web 应用的核心和基础。另外值得注意的是，虽然称为前端、后端，但两者都是部署于应用服务器之上的，即使是用户界面，也需要用户浏览器向应用服务器请求后，由服务器传输相应的页面内容至用户浏览器，并由浏览器将其显示给用户查看和操作。不能看到前端内容就认为其内容是直接部署于客户端的浏览器上的。

目前业界用来开发前端内容的应用技术涵盖面较广，从广义上来说，甚至包括图像处理、人机交互设计等方面的技术，但对 Web 应用开发工程师而言，常用的前端应用技术主要包括 HTML 与 JavaScript 等，下面分别简单介绍。

HTML：如前面万维网的定义以及 Web 应用的发展历史中所述，在当前的 WWW 网即万维网上，最基本的单位为一个超文本文档，每个文档则被称为页面（英文称 page），大多数 Web 应用通常会提供一个访问初始页面，称为主页（Homepage）或首页，然后应用中的每个页面都会蕴涵文本、多媒体等各项内容，另外还会包含指向其他相关页面或节点的指针，一般称为超级链接（由 URL 表示）。而所有的 Web 页面都是使用 HTML 技术来设计的。所谓 HTML（Hyper Text Markup Language），即超级文本标记语言，顾名思义，它是一种基于标记语言，而它的设计目的则是用于创建超文本文档，使得这些文档能够在网页浏览器中得到显示，从而向用户提供信息。利用 HTML 语言，开发人员可以使用其所规定的各类标记符号来定义网页的各部分内容及其外观。从本质上说，HTML 文件是基于文本的，但在文本内容之外，HTML 规定了各类标记符，这些标记符被添加到文本内容中，用于设置这些内容与显示相关的各种特性（例如，设置不同的文字字体、文字大小；对文字内容进行分段；设置图片大小及其显示的具体位置等）。而支持 HTML 的浏览器则对整个文件的所有内容进行顺序读取，并将标记和文本内容进行区分处理，最终实现按标记符的定义和设置显示其所标记的内容，使得用户能够从浏览器中查看到相应内容信息。尤其需要注意的是，HTML 语言有其语法规范，然而和普通编程语言相比，基于 HTML 实现的页面中一旦出现了语法方面的错误，浏览器并不会向用户报错并中断处理，而是采取不执行并跳过存在语法错误的 HTML 标记，继续执行后续的页面代码的做法，因此开发者往往不能直接从浏览器中获取语法错误的反馈信息，而只能通过观察页面在浏览器中的显示效果来间接的分析出错原因并进行调试。另外，目前市面上存在不同开发商设计实现的浏览器，它们对同一标记符的处理方式可能存在差异，这时往往会导致相同的页面代码在不同类型的浏览器中有着不同的显示效果。

JavaScript：HTML 构建的页面通常只是包含静态内容的页面，在 Web 应用普及的初期尚能满足人们的需求，随着应用复杂程度的提高，很多应用开始需要与用户进行较为复杂的互动，例如，显示动态的文本内容，对用户浏览器的各类操作进行响应，对 HTML 的元素进行读写以实现动态响应，对用户提交的数据进行验证等，为此人们提出了可在浏览器端执