

## Web应用程序开发基础

本章将对 Web 开发的基础理论、MVC 模式的设计思想和本书所采用的项目进行介绍。通过本章的学习,可以达到以下目标:

- 了解 Web 的发展过程;
- 了解 Web 开发的主要技术;
- 理解 MVC 设计思想。

### 1.1 Web 开发概述

#### 1.1.1 Web 技术的发展

随着 Internet 技术的发展,Web 技术已经被广泛地应用于 Internet 上,但早期的 Web 应用全部是静态的 HTML 页面,可以将一些文本信息呈现给浏览者,但 HTML 页面中的内容是固定不变的,因此不具备与用户交互的能力,没有动态显示功能。

很自然地,人们希望 Web 应用中应该包含一些能动态执行的页面,最早的 CGI (Common Gateway Interface,通用网关接口)技术满足了该要求,CGI 技术使得 Web 应用可以与客户端浏览器交互,不再需要使用静态的 HTML 页面。利用 CGI 技术可以从数据库中读取信息,将这些信息呈现给用户;还可以获取用户的请求参数,并将这些参数保存到数据库中。

CGI 技术开启了动态 Web 应用的时代,给了这种技术无限的可能性。但 CGI 技术存在很多缺点,其中最大的缺点就是开发动态 Web 应用难度非常大,而且在性能等各方面也存在限制。至 1997 年,随着 Java 语言被广泛使用,Servlet 技术迅速成为动态 Web 应用的主要开发技术。相比传统的 CGI 应用而言,Servlet 具有很大的优势。

Servlet 在 Web 应用中被映射成一个 URL (Uniform Resource Locator,统一资源定位符),该 URL 可以被客户端浏览器请求,当用户向指定 URL 对应的 Servlet 发送请求时,该请求将被 Web 服务器接收到,由 Web 服务器负责处理多线程、网络通信等工作,而 Servlet 的内容则决定了服务器对客户端的响应内容。

到了 1998 年,微软发布了 ASP 2.0。它是 Windows NT 4 Option Pack 的一部分,作

为 IIS 4.0 的外接式附件。它与 ASP 1.0 的主要区别在于它的外部组件是可以初始化的,这样,ASP 程序内部的所有组件都有了独立的内存空间,并且可以进行事务处理,这标志着 ASP 技术开始真正作为动态 Web 编程技术。

当 ASP 技术在世界上广泛流行时,人们很快感受到这种简单技术的魅力:ASP 使用 VBScript 作为脚本语言,它的语法简单、开发效率非常高。而且,世界上已经有了非常多的 Visual Basic 程序员,这些 Visual Basic 程序员可以很轻易地过渡成为 ASP 程序员。因此,ASP 技术迅速成为应用最广泛的动态 Web 开发技术。

随后,由 Sun 带领的 Java 阵营立即发布了 JSP 标准,从某种程度上来看,JSP 是 Java 阵营为了对抗 ASP 推出的一种动态 Web 编程技术。

ASP 和 JSP 的名称很相似,但它们的运行机制存在一些差别,这主要是因为 VBScript 是一种脚本语言,无须编译,而 JSP 使用 Java 作为脚本语句,但 Java 从来就不是解释型的脚本语言,因此 JSP 页面并不能立即执行。因此,JSP 必须编译成 Servlet,也就是说,JSP 的实质还是 Servlet。不过,编写 JSP 比编写 Servlet 要简单得多。

随着实际 Web 应用的使用越来越广泛,Web 应用的规模也越来越大,开发人员发现动态 Web 应用的维护成本也越来越大,即使只需修改该页面的一个简单按钮文本,或者一段静态的文本内容,也不得不打开混杂的动态脚本页面源文件进行修改,这是一种很大的风险,完全有可能引入新的错误。

这时候,人们意识到,使用单纯的 ASP 或者 JSP 页面充当过多角色是相当失败的选择,这对于后期的维护相当不利。慢慢地开发人员开始在 Web 开发中使用 MVC 模式。

随后 Java 阵营发布了一套完整的企业开发规范:J2EE(现在已经更名为 Java EE),紧跟着微软也发布了 ASP.NET 技术,它们都采用一种优秀的分层思想,力图解决 Web 应用维护困难的问题。

### 1.1.2 静态 Web 和动态 Web 的区别与联系

静态网站是最初的建站方式,浏览者所看到的每个页面是建站者上传到服务器上的一个 .htm 或 .html 文件,对于这种网站每增加、删除、修改一个页面,都必须重新对服务器上的文件进行一次下载和上传操作。网页内容一经发布到网站服务器上,无论是否有用户访问,每个静态网页的内容都保存在网站服务器上,也就是说,静态网页是实实在在保存在服务器上的文件,每个网页都是一个独立的文件。静态网页的内容相对稳定,因此容易被搜索引擎检索到,但存在着以下两个缺点。

(1) 静态网页没有数据库的支持,在网站制作和维护方面工作量较大,因此当网站信息量很大时,完全依靠静态网页制作起来比较困难。

(2) 静态网页的交互性较差,在功能方面有较大的限制。

随着人们对动态效果的需求不断增加,出现了动态网站。这里所说的动态网页并不是指网页上简单的 GIF 动态图片或是 Flash 动画,与网页上的各种动画、滚动字幕等视觉上的“动态效果”没有直接关系。动态网页可以是纯文字内容的,也可以是包含各种动画的内容,这些只是网页具体内容的表现形式,无论网页是否具有动态效果,采用动态网站技术生成的网页都称为动态网页,都具备以下 3 个基本特征。

(1) 交互性：网页会根据用户的要求和选择而动态地改变和响应，浏览器作为客户端，成为一个动态交流的桥梁，动态网页的交互性也是 Web 今后发展的方向。

(2) 自动更新：即无须手动更新 HTML 文档，便会自动生成新页面，可以大大节省工作量。

(3) 因时因人而变：即当不同时间、不同用户访问同一网址时会出现不同页面。

动态网站在页面里嵌套了程序，将一些框架相同、更新较快的信息页面进行了内容与形式的分离，将信息内容以记录的形式存入网站的数据库中，以便网站各处的调用。这样用户看到的页面可能和服务上 HTML 文件就不一一对应了，网页框架里调用了很多数据库记录中的内容。此外动态网页是与静态网页相对应的，也就是说，网页文件的扩展名不是 .htm、.html、.shtml、.xml 等静态网页的常见形式，而是以 .asp、.jsp、.php、.perl、.cgi 等为扩展名。

从网站浏览者的角度来看，无论是动态网页还是静态网页，都可以展示基本的文字和图片信息，但从网站开发、管理、维护的角度来看有很大的差别。静态 Web 无法进行数据库操作，而动态 Web 是可以进行数据库操作的。现在几乎所有的数据都是通过数据库来保存的，也正是由于这个原因，动态 Web 开发已经被广泛应用到各个行业之中。

### 1.1.3 Web 应用系统的开发模式

Web 应用系统有两种模式：C/S 模式和 B/S 模式。

(1) C/S(Client/Server, 客户端/服务器端)模式：这种模式功能强大，一般对环境要求比较高，实时交互性好，对于开发而言比较复杂，维护比较麻烦，需要分别安装客户端和服务端。例如，人们日常生活中使用的 QQ 或 MSN 等，都属于 C/S 模式。

(2) B/S(Browser/Server, 浏览器/服务器)模式：相当于在 C/S 模式中以浏览器作为客户端的情况。只是在一般情况下操作系统已经集成了这个客户端，因而不用再安装了，B/S 模式结构比较简单，维护比较方便，只需管理和维护服务器端即可。例如，购物网站或论坛都采用 B/S 模式。

### 1.1.4 Model 1 和 Model 2

对于动态 Web 编程技术而言，则经历了所谓的 Model 1 和 Model 2 时代。

所谓 Model 1 就是 JSP 大行其道的时代。在 Model 1 模式下，整个 Web 应用几乎全部由 JSP 页面组成，由 JSP 页面接收并处理客户端请求，之后直接做出响应。用少量的 JavaBean 来处理数据库连接、数据库访问等操作。图 1-1 显示了 Model 1 的程序流程。

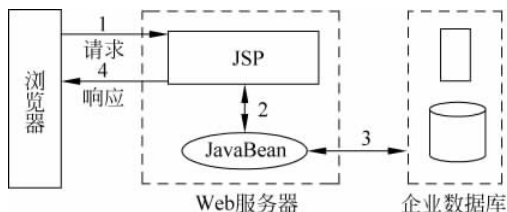


图 1-1 Model 1 的程序流程

Model 1 模式的实现比较简单,适用于快速开发小规模项目。但从工程化的角度来看,它的局限性非常明显:JSP 页面身兼 View 和 Controller 两种角色,将控制逻辑和表现逻辑混杂在一起,从而导致代码的重用性非常低,减小了应用的扩展性,增加了维护的难度。

早期有采用大量 JSP 技术开发的 Web 应用,这些 Web 应用都采用了 Model 1 架构。

Model 2 已经是基于 MVC 架构的设计模式。首先,在 Model 2 架构中,Servlet 作为前端控制器,负责接收客户端发送的请求,在 Servlet 中只包含控制逻辑和简单的前端处理程序;其次,调用后端 JavaBean 来完成实际的逻辑处理;最后,转到相应的 JSP 页面处理显示逻辑。其具体实现方式如图 1-2 所示。

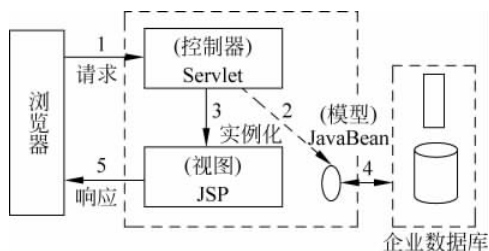


图 1-2 Model 2 的程序流程

从图 1-2 中可以看到,在 Model 2 下 JSP 不再承担控制器的责任,它只在表现层发挥作用,仅用于将结果呈现给用户,JSP 页面的请求将与 Servlet(控制器)交互,而 Servlet 负责与后台的 JavaBean 通信。在 Model 2 模式下,模型(Model)由 JavaBean 充当,视图(View)由 JSP 页面充当,而控制器(Controller)则由 Servlet 充当。

由于引入了 MVC 模式,使 Model 2 具有组件化的特点,更适用于大规模应用的开发,但也增加了应用开发的复杂程度。原本需要一个简单的 JSP 页面就能实现的应用,在 Model 2 中被分解成多个协同工作的部分,需要花更多时间才能真正掌握其设计和实现过程。

Model 2 已经是 MVC 设计思想下的架构,下面简要介绍 MVC 设计思想。

## 1.2 MVC 设计思想

Java EE (Java Enterprise Edition)是在 Java SE 基础上建立起来的一种标准开发架构,主要应用于企业级应用程序的开发。在 Java EE 的开发中以 B/S 模式作为主要的开发模式。在整个 Java EE 中最核心的设计模式就是 MVC(Model-View-Controller)设计模式,并且被广泛应用。M 指模型,V 指视图,C 指控制器。引入 MVC 模式的目的是实现 Web 系统的职能分工。模型用于实现系统中的业务逻辑,通常可以用 JavaBean 或 EJB 来实现。视图用于实现与用户的交互,通常用 JSP 来实现。控制器层是模型与视图之间沟通的桥梁,它可以分派用户的请求并选择恰当的视图以用于显示,同时它也可以解释用户的输入并将它们映射为模型可执行的操作。

MVC 并不是 Java 语言所特有的设计思想,也并不是 Web 应用所特有的思想,它是

所有面向对象程序设计语言都应该遵守的规范。

MVC 思想将一个应用分成 3 个基本部分：模型、视图和控制器，这 3 个部分以最小的耦合协同工作，从而能够提高应用的可扩展性及可维护性。

起初，MVC 模式是针对相同的数据需要不同显示的应用而设计的，其整体结构如图 1-3 所示。

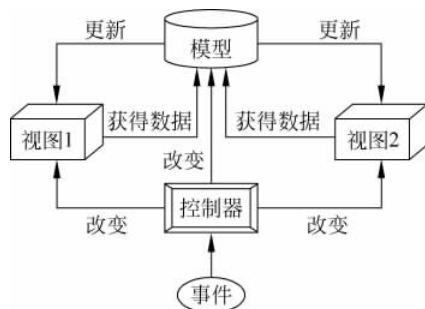


图 1-3 MVC 结构

在经典的 MVC 模式中，事件由控制器处理，控制器根据事件的类型改变模型或视图；反之亦然。具体地说，每个模型对应一系列的视图列表，这种对应关系通常采用注册的方法来完成，即把多个视图注册到同一个模型，当模型发生改变时，模型向所有注册过的视图发送通知，接下来，视图从对应的模型中获得信息，然后完成视图显示的更新。

从设计模式的角度来看，MVC 思想非常类似于一个观察者模式，但与观察者模式存在少许差别：在观察者模式下观察者和被观察者可以是两个互相对等的对象，但对于 MVC 思想而言，被观察者往往只是单纯的数据体，而观察者则是单纯的视图页面。

概括起来，MVC 有如下特点。

(1) 多个视图可以对应一个模型。按 MVC 设计模式，一个模型对应多个视图，可以减少复制及维护的代码量，一旦模型发生改变，也易于维护。

(2) 模型返回的数据与显示逻辑分离。模型数据可以应用任何的显示技术，例如，使用 JSP 页面、Velocity 模板或者直接产生 Excel 文档等。

(3) 应用被分隔为 3 层，减小了各层之间的耦合，提高了应用的可扩展性。

(4) 控制层的概念也很有效，由于它把不同的模型和不同的视图组合在一起来完成不同的请求，因此，控制层可以说是包含了用户请求权限的概念。

(5) MVC 更符合软件工程化管理的精神。不同的层各司其职，每一层的组件具有相同的特征，有利于通过工程化和工具化产生管理程序代码。

相对于早期的 MVC 思想，Web 模式下的 MVC 思想则又存在一些变化，因为对于一个应用程序而言，可以将视图注册给模型，当模型数据发生改变时，及时通知视图页面发生改变；而对于 Web 应用而言，即使将多个 JSP 页面注册给一个模型，当模型发生变化时，模型无法主动发送消息给 JSP 页面（因为 Web 应用都是基于请求/响应模式的），只有当用户请求浏览该页面时，控制器才负责调用模型数据来更新 JSP 页面。

## 1.3 项目概述

本书所采用的项目是某环境规划院的人事管理系统。人事管理是企业生存的主要因素,人员的增减、变动将直接影响到企业的整体运作,企业每天都要涉及人员管理问题。企业员工越多、分工越细、联系越紧密,所要做的统计工作就越多,人事管理的难度就越大。本书完成的系统为某环境规划院人事管理系统,该系统基本上能满足现代企业人事管理的需求,人事管理系统中保存了相关的人员信息,方便查询、浏览、修改等操作。

本系统的使用者为某环境规划院的管理人员。通过该系统用户能够对系统进行系统设置、员工资料、人事资料的管理,而且可以对各种资料进行统计查询。

本系统具有的功能包括系统设置、员工管理、人事管理、统计管理、工资管理、休假管理。每个功能的具体描述如下。

### 1. 系统设置

(1) 部门管理:对部门信息进行设置。

(2) 职位管理:对职位信息进行设置,包括的职位有办公室主任、院长、总工、副院长等。

(3) 职称管理:对职称信息进行设置,包括的职称有工程师、教授等。

(4) 奖惩信息设置:对奖惩信息进行设置,如迟到、早退、旷工等。

(5) 用户管理:对使用该系统的人员信息进行设置。

### 2. 员工管理

员工管理主要是对院内员工的基本信息进行管理,主要包括以下几部分的功能。

(1) 员工基本信息录入:录员工的个人信息。

(2) 员工信息编辑:编辑修改员工信息。

(3) 员工信息删除:删除不在本单位工作的员工信息(为了保证信息的完整性,此处的删除功能为逻辑删除)。

(4) 模糊查询员工信息:根据输入的模糊查询条件,查询出所需要的员工信息。

(5) 员工信息打印:打印员工的个人信息。

(6) 员工相关资料管理:对员工的业绩考核资料及其他资料进行管理 & 维护操作。

(7) 员工基本信息的导入与导出:用于导入和导出员工信息,可以直接进行打印或保存为 Excel 文件。

(8) 员工状态操作:如实习、转正、退休等。

(9) 员工附加信息、简历等管理。

(10) 组织撤销:相应组织被撤销后仍保留历史薪资统计数据。

(11) 岗位调动时自动更新上下级关系(可使用拖拉操作)。

(12) 可快速查询各组织内职位和人员配备情况。

(13) 员工证件管理。

(14) 合约到期和试用期到期提醒功能。

(15) 可预先设定员工离职日期,系统自动处理离职信息。

(16) 内建离职员工数据库,保留离职员工的所有数据。

### 3. 人事管理

- (1) 奖惩资料管理。
- (2) 培训资料管理。
- (3) 考评资料管理。
- (4) 调动资料管理。
- (5) 工资资料管理。
- (6) 其他信息。

在以上的功能模块可以进行相应信息的添加、修改、删除及查询操作。

### 4. 统计管理

(1) 按条件进行统计,如按部门统计、按职称统计等,统计的内容主要有平均年龄,职称结构,学历结构,平均工资,某一部门的人数,某一职称、某一学历的人数。

(2) 员工生日统计:统计某一月份或时间段内的员工生日,以便领导发送生日祝福等,提高员工的工作积极性及热情。

(3) 外聘人员与正式人员统计:统计各部门正式在编人员、外聘人员及学生的情况。按时间段进行统计。

(4) 人事信息统计:按姓名、性别、职务、学历等进行统计,可加入条件显示人员信息。

(5) 人事记录统计:按员工、部门、月份、项目等统计培训、调薪、调动等情况,以统计图的形式进行显示。

### 5. 工资管理

本系统预留与财务 NC 系统的接口,可将工资信息导入内网办公系统中,提供工资信息,由员工进行查询。

### 6. 休假管理

- (1) 支持多种休假政策。
- (2) 支持多种假期表。
- (3) 支持日和小时两种休假时间单位。
- (4) 支持多种休假类型(如年假、病假等)。
- (5) 支持加班转调休。
- (6) 可输入或者导入员工休假或者加班的明细数据。
- (7) 可查询员工休假或者加班的明细数据。
- (8) 可汇总统计休假或者加班数据。

本书以该人事管理系统中部分模块的开发为需求背景,介绍当前流行的三大框架技术,其中包括:使用 Struts 2 技术完成职称系列管理模块的开发,使用 Struts 2+Hibernate 框架技术完成用户管理模块的开发,使用 Struts 2+Hibernate+Spring 技术完成部门管理模块的开发,最后的一章将详细列出证件信息管理模块的需求及设计说明,以分层架构的形式给出实现证件信息管理模块的关键代码。

## 小结

本章对 Web 基础理论进行了介绍,包括 Web 技术的发展、静态 Web 和动态 Web 的区别与联系、C/S 模式和 B/S 模式,之后介绍了 Model 1 和 Model 2 的简要模型和特征,进而介绍了 MVC 模式的主要策略和主要优势。最后对本书所采用的项目需求进行了描述。

## 习题

### 简答题

1. 比较静态 Web 和动态 Web 的区别与联系。
2. 怎样理解 C/S 模式和 B/S 模式?
3. 比较 Model 1 和 Model 2,指出它们各自的优缺点。
4. 简述 MVC 设计模式的组成部分。
5. 简述如何理解 MVC。

# 搭建运行和开发环境

进行系统开发前需要做一些前期准备,本章将对运行环境和开发环境的搭建进行说明。通过本章的学习,可以达到以下目标:

- 学会 JDK 的安装和环境变量配置;
- 掌握 Eclipse 和 MyEclipse 的概念及安装过程;
- 能够在 Eclipse 中创建 Web 应用程序;
- 掌握 Web 服务器的安装及使用方法。

## 2.1 搭建运行环境

### 2.1.1 下载并安装 JDK

JDK(Java Development Kit)是 Sun 公司提供的—个开源、免费的 Java 开发工具。JDK 是整个 Java 的核心,包括 Java 运行环境、Java 工具和 Java 基础类库。在安装其他开发工具和集成开发环境以前,必须首先安装 JDK。可以从 Sun 公司的官方网站获得免费的 JDK。

JDK 下载完成之后,按照提示即可完成安装。

### 2.1.2 Java 环境变量的设置

JDK 安装完成后,需要设置环境变量,Windows 操作系统下的环境变量配置步骤如下。

(1) 右击“我的电脑”图标,在弹出的快捷菜单中选择“属性”菜单项,打开“系统属性”对话框,选择“高级”选项卡,单击“环境变量”按钮,打开“环境变量”对话框。

(2) 配置用户变量。

① 新建 JAVA\_HOME 变量,输入“C:\Program Files\Java\jdk1.6.0(JDK 的安装路径)”。

② 新建 PATH 变量,输入“;%JAVA\_HOME%\bin; %JAVA\_HOME%\jre\bin;”。

③ 新建 CLASSPATH 变量,输入“.;%JAVA\_HOME%\lib; %JAVA\_HOME%\lib\tools.jar”。

(3) 测试环境变量配置是否成功。选择“开始”→“运行”菜单项,打开“运行”对话框,

在“打开”文本框中输入“cmd”命令,单击“确定”按钮打开命令提示符窗口。在命令提示符窗口中,分别输入命令“java”(java.exe是指Java解释器,用来解释执行字节码文件)和“javac”(javac.exe是指Java编译器,用来编译源文件,得到字节码文件),并按Enter键,此时命令提示符窗口中出现相应的命令,而不是出错信息,即表示配置成功。

### 2.1.3 Tomcat 安装和配置

如果进行的是Java Web的开发,还需要安装Web服务器, Tomcat服务器不仅是一个Servlet容器,也是一个免费的开放源代码的Web应用服务器,它是Apache软件基金会(Apache Software Foundation)Jakarta项目中的一个核心项目,由Apache、Sun和其他一些公司及个人共同开发而成。因为Tomcat技术先进、性能稳定,而且免费,因而深受Java爱好者的喜爱并得到了部分软件开发商的认可,成为目前比较流行的Web应用服务器。

可以直接从Apache的网站上下载Tomcat。在安装Tomcat之前要先安装JDK,本书使用的Tomcat版本为Tomcat 6.0.18,需要安装J2SE 5.0(JDK 1.5)以上的版本才能运行。对于可执行的安装程序,只需双击这个文件,就可以开始安装了。本书将Tomcat 6.0服务器安装在了C盘的Tomcat 6.0目录中,在安装过程中需要配置服务器的端口号和密码,本书中Tomcat服务器的端口号是8080,密码为空。在安装过程中安装程序会自动搜索JDK和JRE的位置。

服务器安装完成之后,即可通过Tomcat中bin目录下的tomcat6.exe启动Tomcat服务器。

Tomcat启动成功后,打开浏览器,在地址栏中输入“http://localhost:8080/”,可以看到如图2-1所示的界面。如果出现此界面,说明Tomcat安装成功。



图 2-1 Tomcat 安装成功界面