

# 基于 JavaMail 的 Web 邮件客户端的设计与实现

## Design and Implementation of Web Mail Client Based on JavaMail

周红静 ( 湖南商学院 计算机与电子工程学院 湖南长沙 410205 )

**摘要:** 分析了 JavaMail 的体系架构,阐述了在 Windows 平台上 Web 邮件客户端系统的设计方法,并给出了详细的实现过程。该 Web 邮件客户端实现了邮件的接收、发送、管理以及个人通讯录等功能,具有较强的应用价值。

**关键词:** 电子邮件 JavaMail JAF JSP

E-mail( Electronic mail ,电子邮件 )是网络用户最常用的网络服务之一<sup>[1]</sup>。人们收发邮件常用的方式主要有两种。一种是通过专用邮件客户端软件来收发邮件,例如 Outlook、Foxmail。这种方式的缺点是每位用户的计算机上必须安装邮件客户端软件,而且用户接收和整理好的邮件不能在其它机器上直接进行访问。另一种是通过 Web 浏览器来收发邮件。使用这种方式,人们不需要安装任何专用的邮件客户端软件,只要有浏览器,邮件用户可以在任何地方通过它直接登录到邮件服务器来收发邮件,非常简单、方便。本文详细阐述了基于 Sun 公司的 JavaMail<sup>[2]</sup>技术设计和开发一个 Web 邮件客户端软件的方法和过程。

## 1 JavaMail 技术

JavaMail 是 Sun 公司为 Java 开发人员提供的公用应用程序接口框架,提供了一个与平台无关、协议独立的框架来构建邮件和消息系统。JavaMail 体系结构分为三个层次,即抽象类层、实现类层和 JAF ( JavaBeans Activation Framework, JavaBeans 活动框架 )<sup>[3]</sup>,其结构如图 1 所示。

其中抽象类层和实现类层由 JavaMail API 提供,而 JAF 不由 JavaMail API 提供。抽象类层支持邮件处理功能。实现类层实现了部分 Internet 标准,如 RFC822 和 MIME。JAF 用来封装消息数据和处理与数据交互的命令。JavaMail 的层次体系结构允许客户使用相同的 JavaMail API 来发送和存储各种类型的消息,为收发信

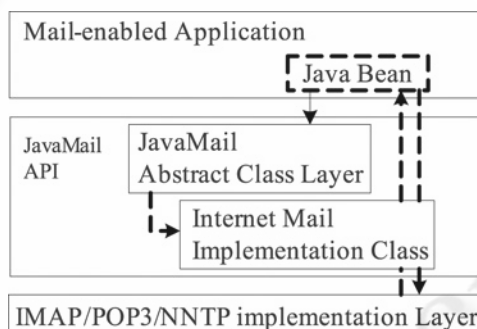


图 1 JavaMail 框架

息提供与协议无关的访问,提高了灵活性和可扩展性。JavaMail API 提供了一个抽象类集合,这些抽象类定义了组成邮件系统对象。同时,这些 API 能够被扩展来提供新的协议和增加功能。此外 API 还提供了具体的子类用于应用程序的开发。

JavaMail 具有下面的一些特点 ( 1 )简单、直接,对于开发者易于实现 ( 2 )采用常见的概念和编程模型支持代码的开发,与其它的 Java API 有良好的接口; ( 3 )轻量级的类和接口使任何应用程序加入基本的邮件处理任务都变得容易 ( 4 )支持复杂的邮件格式、数据类型以及传输协议。

## 2 系统设计

### 2.1 Web 邮件系统架构

目前,许多邮件系统采用多层次架构,按照数据处理功能把系统在逻辑上分成多个层次,具有较好的扩

展性和移植性<sup>[4]</sup>。整个 Web 邮件系统的设计采用三层结构,划分为表示层、业务逻辑层和数据服务层,系统架构如图 2 所示。

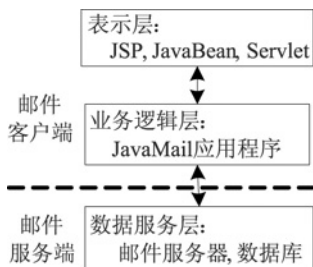


图 2 Web 邮件系统架构

邮件客户端包括表示层和业务逻辑层两个层次。表示层由用户直接通过浏览器访问的 JSP 页面以及 JavaBean 和 Servlet 组成。其中 JavaBean 和 Servlet 用于 JSP 页面与 JavaMail 应用程序之间的逻辑控制。业务逻辑层则由负责发送、接收和管理邮件的 JavaMail 应用程序组成,完成用户提交的任务,是系统的核心应用程序。邮件服务端包含数据服务层,数据服务层用来存储用户邮件以及用户的相关信息。

## 2.2 邮件客户端功能模块

本邮件客户端包括以下五个基本功能模块:登录模块、邮件接收模块、邮件发送模块、邮件管理模块以及通讯录管理模块。功能结构如图 3 所示。



图 3 Web 邮件客户端功能结构

(1)登录模块:实现用户登录,根据用户名和口令验证来连接邮件服务器。

(2)邮件接收模块:包括显示邮件列表和阅读邮件两个部分。接收到的邮件将直接保存在收件箱中,采用邮件列表的形式显示在页面上,每封邮件只显示主题、发件人、发送时间等简单信息,用户可以点击邮

件主题查看邮件的具体内容。每个用户都有一个用户文件夹,程序通过不同的参数来定位文件夹。

(3)邮件发送模块:包括编辑邮件和发送邮件两个部分。编辑邮件这部分主要实现写邮件的功能。在这个系统中,有三种编辑邮件的模式:①编辑新邮件。此模式下进入编辑,创建的是全新的邮件,所有信息都要用户输入。②回复、转发邮件。此模式下进入编辑,邮件的收件人、主题、邮件内容都由原邮件决定,但是可以由用户更改。③编辑草稿。此模式下是直接对已经保存在草稿箱中的邮件进行编辑。发送邮件这部分在邮件编辑状态直接点击“发送”按钮即调用相关的应用程序将邮件发送到目的地。

(4)邮件管理模块:主要实现邮件的删除和移动。可以直接从文件夹中删除邮件,也可以将邮件移动到已有的文件夹或用户自己建立的文件夹,方便邮件的分类存储和阅读。此模块中,对邮箱文件夹进行操作由 JavaMail 中相关的类和方法来实现,并可以提供给其它程序调用。

(5)通讯录管理模块:主要实现用户通讯录记录的添加、删除和维护,利于用户对联系人地址的管理。

## 3 系统详细实现

### 3.1 开发环境

本系统是在 Windows XP 操作系统上进行开发的,支持 JDK 5.0、JavaMail API 和 JAF。Web 服务器采用的是 Tomcat 5.0,数据库为 SQL Server 2005,邮件服务器选用的是 Exchange 2003。

### 3.2 实现方式

Web 邮件客户端从实现角度来分析,包括两个部分:前台和后台。前台实现 JSP 页面、JavaBean 和 Servlet。后台实现 JavaMail 应用程序。其处理过程如图 4 所示。

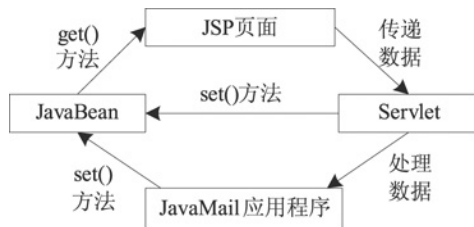


图 4 Web 邮件客户端处理过程

### 3.2.1 前台功能实现

系统前台每个模块都是由 Pagedata、JSP 页面和 Servlet 组成。其中 ,Pagedata 为一个 JavaBean ,它类似于一个数据结构 ,封装了相应页面所需要传递的数据 ,并提供 set( )和 get( )方法设置和读取数据。JSP 页面又由主页面和对应的 data 页面构成 ,data 页面具体定义了主页面所需要的数据 ,主页面只需要将其 data 页面包含进来即可以访问这些数据。Servlet 作为控制器对页面请求进行处理 ,处理完成后通过 Pagedata 的 set( )方法将数据存入 JavaBean ,然后控制页面跳转到的页面再通过 Pagedata 的 get( )方法读取数据并显示出来。

系统中的 Servlet 主要包括以下几个方面的处理。LoginServlet 是对成功登录后页面的控制。当用户输入用户名和密码并且成功通过验证后 ,LoginServlet 将控制页面跳转到 menu 界面 ,并自动取得 menu 中要显示的数据显示在页面上。比如取得个人文件夹的名称和个数显示在 menu 中。FolderServlet 是对邮箱文件夹的处理。用户可以通过页面调用它来取得邮箱的所有文件夹 ,并可显示所有的文件夹及其包含的邮件数。ListMailServlet 是对邮箱文件夹中邮件列表显示、移动邮件、删除邮件、查找邮件、过滤邮件等操作的控制。ViewMailServlet 用于阅读邮件控制。当用户在页面点击某一邮件时 ,此 Servlet 将调用 impl 包中相应方法 ,将此邮件的具体内容显示在另一个 JSP 页面上并从当前页面跳转到此页面。WriteMailServlet 用于发送邮件、保存邮件以及回复邮件控制。AddressServlet 用于通讯录记录处理。

### 3.2.2 后台程序实现

JavaMail 应用程序是整个系统的指挥中心在后台运行。JavaMail 应用程序负责实现页面提交的请求 ,通过 Servlet 来连接页面和应用程序。Servlet 在处理过程中可以调用 JavaMail 应用程序 ,JavaMail 应用程序也可以通过 set( )方法将数据传递给 JavaBean。

接口 RecvMailConnInterf 由类 ImapConnImpl 实现用户登录认证、邮件接收、邮件解析、邮件发送、邮件列表、邮件文件夹管理等功能。下面给出部分主要功能的实现过程。

(1) 邮件的接收。获取邮件服务器上的用户文件夹 ,然后使用根文件夹的 getMessage( )方法获取邮

件 ,以邮件列表的形式显示。

```
Public Message[ ] getMessages( ) throws MailboxFolderExcep-
tion {
    Message[ ] messages = null ;
    if ( ! openFolder( Folder. READ_ ONLY )) return
    null ;
    try { messages = this. roofolder. getMessages
    ( ) ; // 邮件获取
    } catch ( MessagingException me ) {
    throw ( new MailboxFolderException( " 获取邮件失
    败" , null )) ;
    } finally { closeFolder( ) ; }
    return ( messages ) ;
}
```

(2) 邮件解析。当收到邮件后需要对邮件按格式进行解析 ,MessageParse 类实现了邮件的解析 ,应用于邮件的阅读。它详细定义了邮件的主题、收件人、发件人、正文等信息。下面给出了邮件解析的部分实现内容 :

```
public class MessageParse {
    private Message message ;
    public String getFrom( ) ; // 获取 from 域
    public String getCc( ) ; // 获取 cc 域
    public String getSubject( ) ; // 获取 subject 域
    public String getBody( ) ; // 获取邮件正文
    public String getDate( ) ; // 获取日期
    public int getXPriority( ) ; // 获取邮件优先级
    public void saveAttachment ( int num , Output-
    Stream out ) ; // 保存附件
    .....
}
```

(3) 邮件的发送。sendMessage 方法实现了使用 SMTP 协议发送邮件。通过创建与邮件服务器的 SMTP 会话 ,然后在 SMTP 会话上产生和发送邮件。具体实现如下所示 :

```
public boolean sendMessage( SmtplibMail msgp ) {
    Session session = getSmtplibSession( ) ; // 创建会话
    Message msg = msgp. createMessage( session ) ;
    try { // 发送邮件
        SMTPTransport t = initSMTPTransport( session ) ;
```

```

t.sendMessage( msg ,msg.getAllRecipients( ) );
t.close( );
return true ;
} catch ( MessagingException e ) {
e.printStackTrace( ); }
return false ;
}

```

### 3.3 系统运行示例

经测试系统各功能运行正常。界面设计简洁,方便使用。系统运行时收件箱的界面如图 5 所示。

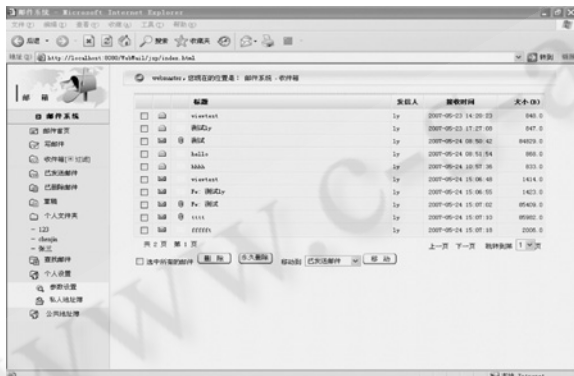


图 5 Web 邮件客户端运行界面示例

## 4 结束语

基于 JavaMail 开发的 Web 邮件客户端系统具有良好的可扩展性,能够支持各种复杂的邮件格式和数据类型。该邮件客户端系统实现了用户登录、邮件发送、邮件接收、邮件管理和通信录管理等基本功能,适用于为企业内部提供邮件服务,具有较强的应用价值。系统进一步的工作可以从以下几个方面展开:增加系统个性化设置模块,满足用户对个性化邮箱的需求。邮件系统安全性控制,包括具备防病毒模块和垃圾邮件过滤模块,以控制病毒和垃圾邮件的传播。

## 参考文献

- 1 中国互联网络信息中心。第 20 次中国互联网络发展状况统计报告. 2007-07-18. <http://www.cnnic.cn/index/0E/00/11/index.htm>.
- 2 Sun Microsystems, Inc. JavaMail API Design Specification Version 1.4. 2007-10-22. <http://java.sun.com/products/javamail/>.
- 3 Sun Microsystems, Inc. JavaBeans Activation Framework Specification Version 1.1. 2006-04-25. <http://java.sun.com/products/javabeans/jaf/index.jsp>.
- 4 钟路,刘玲,夏红霞. 基于 Java Mail API 的 Web 邮件系统开发. 武汉理工大学学报, 2006, 28(6): 84-86.