



开发Java Mail Web应用

- n 课程内容
 - n E-mail协议简介
 - n SMTP简单邮件传送协议
 - n POP3邮局协议
 - n E-mail接收的新协议IMAP
 - n Java Mail API简介
 - n JavaMail应用程序开发环境
 - n 获得JavaMail JAR文件
 - n 安装和配置Mail服务器
 - n 创建Java Mail应用程序
 - n 创建Java Mail Web应用

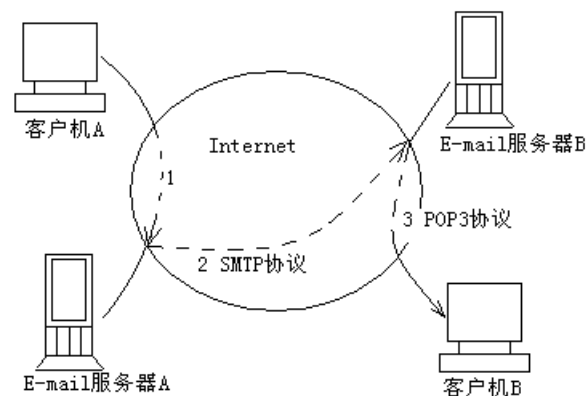
E-mail协议简介

n 发送邮件服务器

- n 使用SMTP协议: Simple Mail Transfer Protocol, 即简单邮件传输协议, 是Internet传送E-mail的基本协议, 也是TCP/IP协议组的成员。SMTP协议解决E-mail系统如何通过一条链路, 把邮件从一台机器传递到另一台机器上的问题。

n 接收邮件服务器

- n 使用POP3或IMAP协议





E-mail协议简介

- n POP3 (Post Office Protocol 3) 即邮局协议第3版，这是 Internet接收E-mail的基本协议，也是TCP/IP协议组的成员。
- n 基于POP3协议的E-mail系统阅读信件的过程如下：

用户通过自己所熟悉的E-mail客户端软件，例如Foxmail、Outlook Express和the Bat等，经过相应的参数设置（主要是设置POP3 邮件服务器的IP地址或者域名、用户帐号及其对应密码）后，只要选择接收邮件操作，就能够将所有的邮件从远端的邮件服务器下载到用户的本地硬盘里。邮件下载之后，用户就可以在本地阅读了。当然，用户如果想节省上网费用，也可以选择脱机状态下慢慢地阅读邮件。

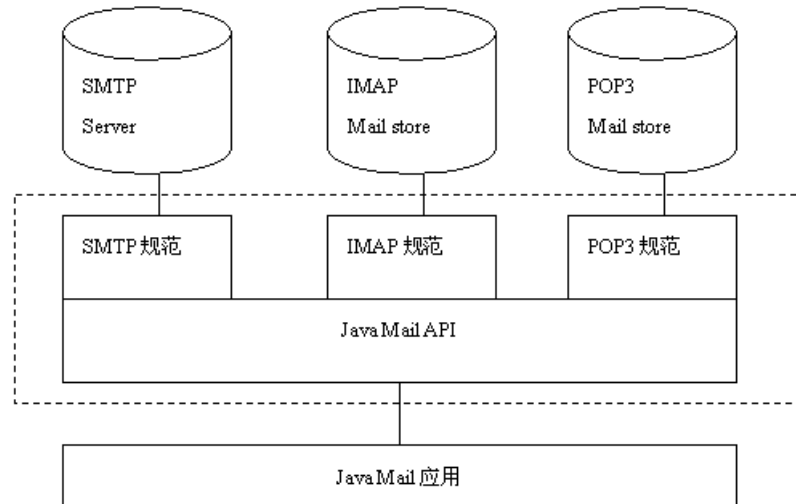


E-mail协议简介

- n **IMAP** (Internet Message Access Protocol)，即直接收取邮件的协议，是与**POP3**不同的一种**E-mail**接收的新协议。**IMAP**协议可以让用户远程访问邮件服务器，并且可以在下载邮件之前预览信件主题与信件来源。有了**IMAP**，用户阅读邮件服务器上的邮件时，就像这些邮件存储在本地机上一样方便。
- n 因为**IMAP**具有远程访问的能力，所以它最有可能被那些认为漫游是重要特性的公司用户所采用。在多数情况下，漫游用户愿意把他们的信件保存在邮件服务器上，这样通过任何一台机器的浏览器都可以收取新的信件或查看旧信。
- n **IMAP**协议不同于**POP3**协议，**POP3**协议将信件存储在一台服务器上，一旦用户和服务器连通，它便将信件发送到客户机上，并从服务器上删除这些邮件。而且，**POP3**不支持用户在服务器上创建邮件夹，可见，使用**POP3**时，用户对在邮件服务器上邮件的控制权很小。

Java Mail API

- n Java Mail API支持各种电子邮件通信协议，如IMAP、POP3和SMTP，为Java应用程序提供了电子邮件处理的公共接口。





Java Mail API的最主要的类

- n javax.mail.Session: 代表邮件会话
- n javax.mail.Store: 代表邮件服务器上的邮件账户的入口
- n javax.mail.Folder: 代表邮件夹
- n javax.mail.Message: 代表电子邮件
- n javax.mail.Address: 邮件地址
- n javax.mail.Transport: 负责发送邮件



Java Mail API的最主要的类

- n **javax.mail.Session**: **Session** 类定义了一个基本邮件会话，是Java Mail API最高层入口类。所有其它类都是经由这个**Session** 才得以生效。**Session** 对象从**java.util.Properties** 对象中获取信息，如邮件发送服务器、接受邮件协议、发送邮件协议、用户名、密码及整个应用程序中共享的其它信息。
- n **javax.mail.Store**: **Store**类是访问接收邮件服务器上邮件账户的入口，通过**Store**类的**getFolder**方法，可以访问特定的邮件夹。
- n **javax.mail.Folder**: **Folder**类代表了邮件夹，用于分级组织邮件，通过**Folder**类可以访问邮件夹中的邮件。



Java Mail API的最主要的类

- n **javax.mail.Message**: **Message**代表了电子邮件。**Message**类封装了邮件信息，提供了访问和设置邮件内容的方法。邮件中包含如下内容：
 - n 地址信息，包括发件人地址、收件人地址列表、抄送地址列表和广播地址列表
 - n 邮件标题
 - n 邮件发送和接收日期
 - n 邮件具体内容
- n **Message** 是个抽象类，常用的子类为 **Javax.mail.internet.MimeMessage**。**MimeMessage** 能支持**MIME** 类型电子邮件消息。**MIME**（**Multipurpose Internet Mail Extensions**）是一种电子邮件编码方式，它可以将发信人的电子邮件中的文本以及各种附件都打包后发送，传送时即时编码，收信人的软件收到邮件后也是即时解码还原，完全自动化，非常方便。当然先决条件是双方的软件都必须支持**MIME**编码，要不然发信人很方便地把信送出去了，但收信人的软件如果没有这种功能，无法把它还原，看到的也就是一大堆乱码了。



Java Mail API的最主要的类

- n `javax.mail.Address`: `Address` 代表了邮件地址，和 `Message` 一样，`Address` 也是个抽象类。常用的子类为 `javax.mail.internet.InternetAddress` 类。
- n `javax.mail.Transport`: `Transport` 类根据指定的邮件发送协议（通常是 `SMTP`），通过指定的邮件发送服务器来发送邮件。`Transport` 类是抽象类，它提供了一个静态方法 `send(Message)` 来发送邮件。

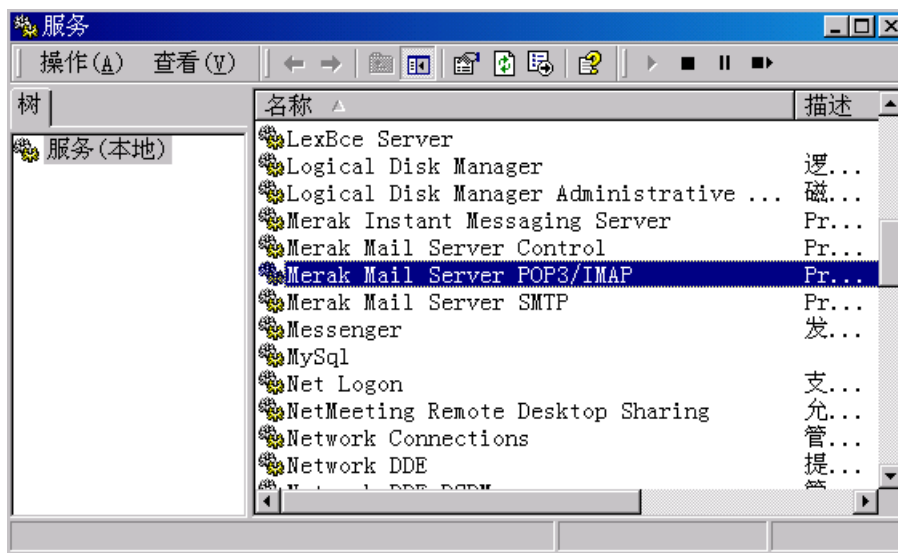


获得JavaMail JAR文件

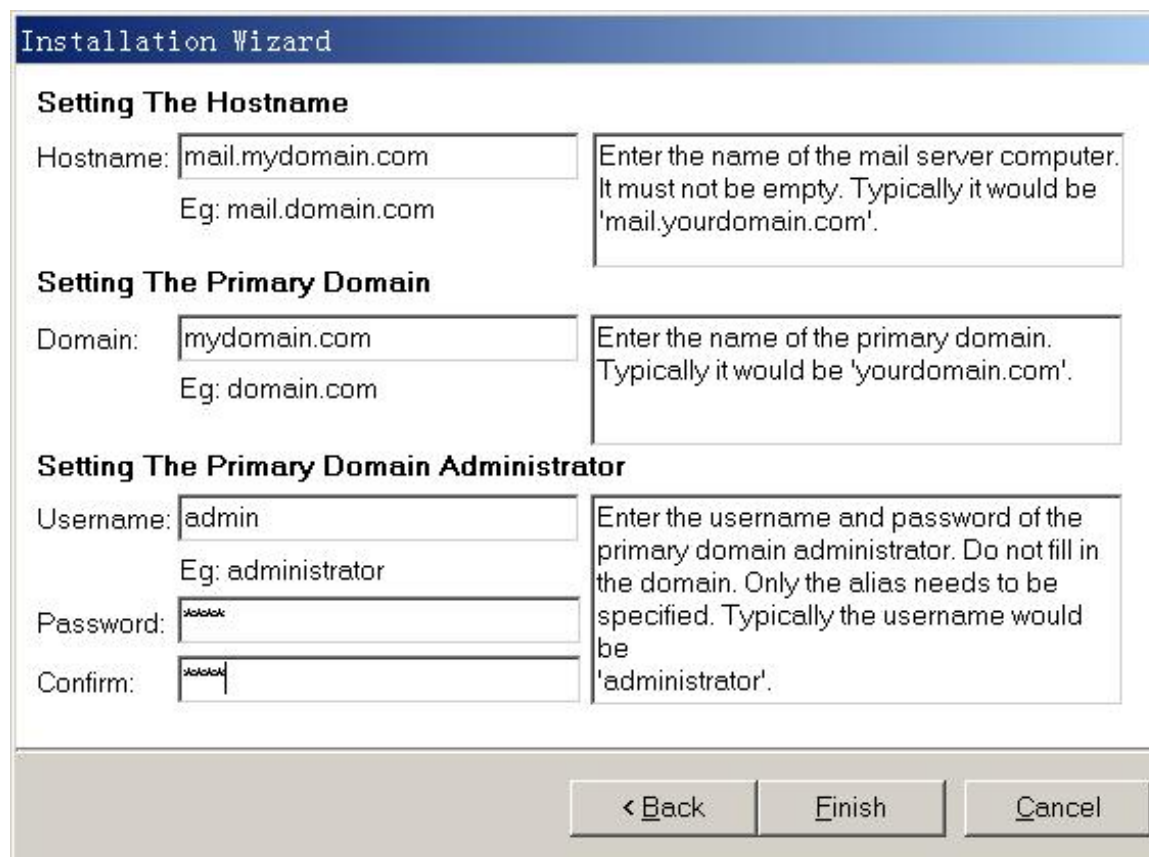
- n 开发JavaMail应用程序需要两个JAR文件: mail.jar和activation.jar, 它包含了JavaMail API中所有的接口和类。
- n 下载地址为:
<http://java.sun.com/products/javamail>

安装MerakMailServer服务器

- n MerakMail服务器安装软件会自动在Window操作系统中加入邮件发送和接收服务，发送邮件采用**SMTP**协议，接收邮件支持**POP3**和**IMAP**协议。每次启动操作系统时，会自动运行这两项服务。



安装过程中配置Domain



Installation Wizard

Setting The Hostname

Hostname:
Eg: mail.domain.com

Enter the name of the mail server computer. It must not be empty. Typically it would be 'mail.yourdomain.com'.

Setting The Primary Domain

Domain:
Eg: domain.com

Enter the name of the primary domain. Typically it would be 'yourdomain.com'.

Setting The Primary Domain Administrator

Username:
Eg: administrator

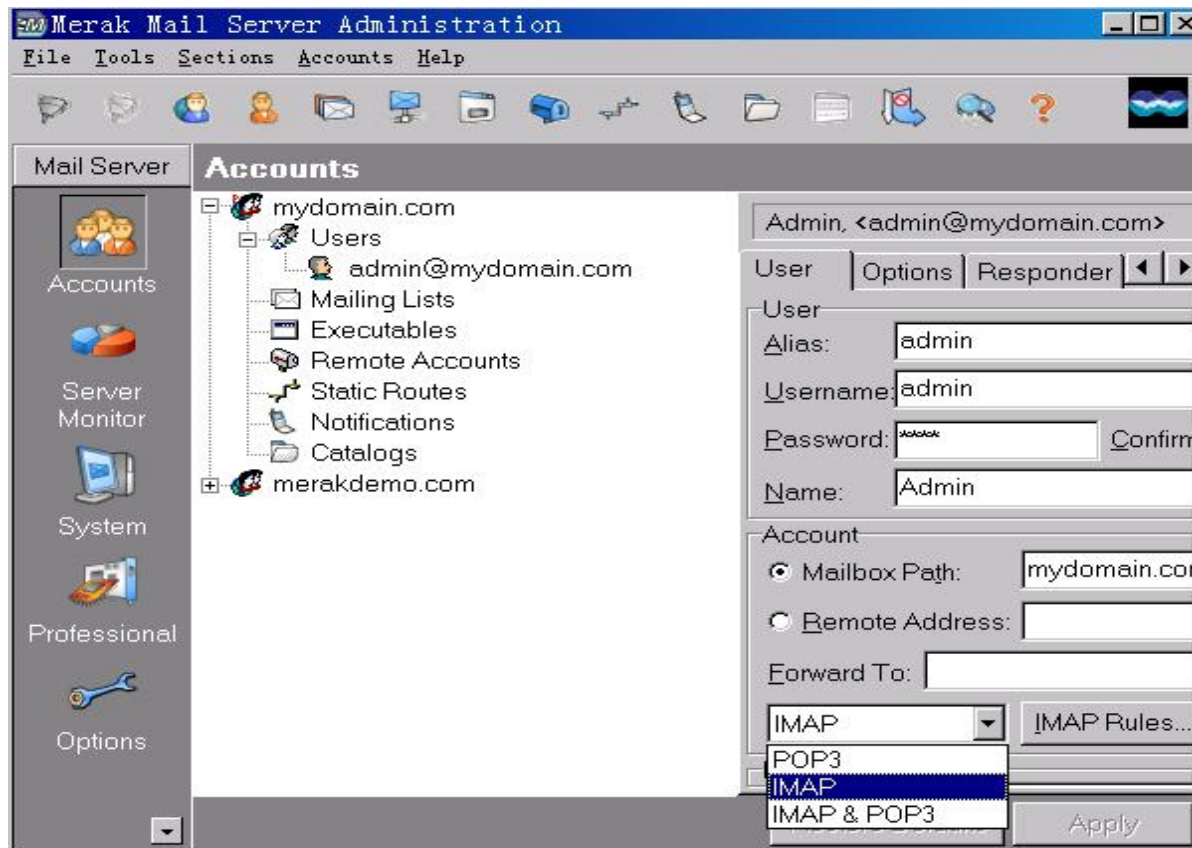
Password:

Confirm:

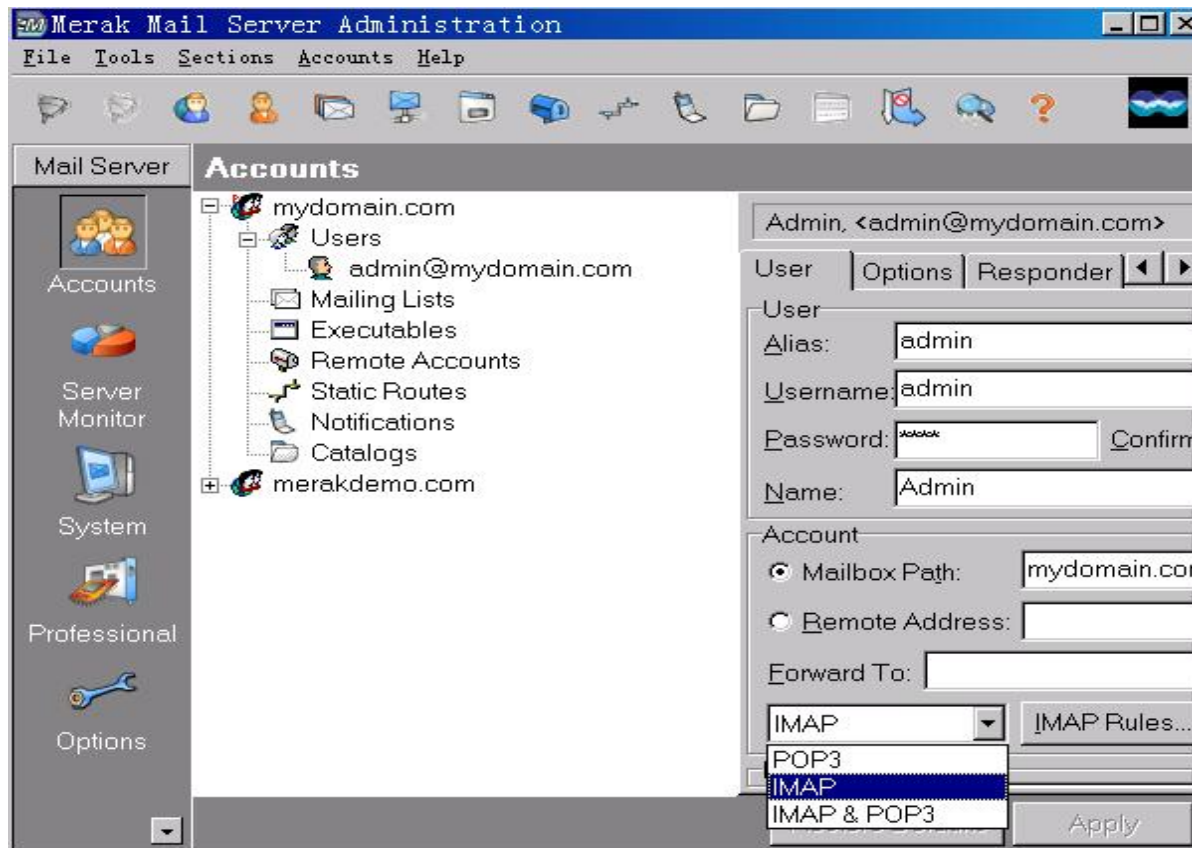
Enter the username and password of the primary domain administrator. Do not fill in the domain. Only the alias needs to be specified. Typically the username would be 'administrator'.

< Back Finish Cancel

MerakMail服务器的管理窗口



MerakMail服务器的管理窗口





创建Java Mail应用程序的初始化步骤

n (1) 设置Java Mail属性:

```
//set properties  
Properties props = System.getProperties();  
props.put("mail.transport.protocol", "smtp");  
props.put("mail.store.protocol", "imap");  
props.put("mail.smtp.class",  
    "com.sun.mail.smtp.SMTPTransport");  
props.put("mail.imap.class",  
    "com.sun.mail.imap.IMAPStore");  
props.put("mail.smtp.host", hostname);
```



创建Java Mail应用程序的初始化步骤

- n (2) 调用 `javax.mail.Session` 类的静态方法 `Session.getDefaultInstance` 获取 `Session` 实例，该方法将根据已经配置的Java Mail属性来创建 `Session` 实例：

```
Session mailsession = Session.getDefaultInstance(props, null);
```

- n (3) 调用 `Session` 的 `getStore` 方法来取得 `Store` 对象，在本例中，将获取采用imap协议的 `Store` 对象：

```
Store store = mailsession.getStore("imap");
```

根据配置的 `mail.imap.class` 属性，可以确定这里 `getStore` 方法返回 `com.sun.mail.imap.IMAPStore` 类型的 `Store` 对象。



创建Java Mail应用程序的初始化步骤

- n (4) 调用Store的connect方法连接到接收邮件服务器上的特定邮件账号。调用connect方法时你应该指定接收邮件服务器名称或IP地址、邮件账户名和口令。

```
store.connect(hostname,username, password);
```



访问邮件服务器上的特定邮件账号

n 创建并发送邮件

参见MailTest.java

```
// create a message
msg = new MimeMessage(mailsession);
InternetAddress[]                                toAddrs
    =InternetAddress.parse("admin@mydomain.com", false);
msg.setRecipients(Message.RecipientType.TO, toAddrs);
msg.setSubject("hello");
msg.setFrom(new InternetAddress("admin@mydomain.com"));
msg.setText("How are you");
//send a message
Transport.send(msg);
```



访问邮件服务器上的特定邮件账号

n 打开inbox邮件夹参看邮件信息:

参见MailTest.java

```
//check inbox
```

```
Folder folder=store.getFolder("inbox");
```

```
folder.open(Folder.READ_WRITE);
```

```
System.out.println("You have "+folder.getMessageCount()+"  
messages in inbox.");
```

```
System.out.println("You have  
"+folder.getUnreadMessageCount()+" unread messages in  
inbox.");
```



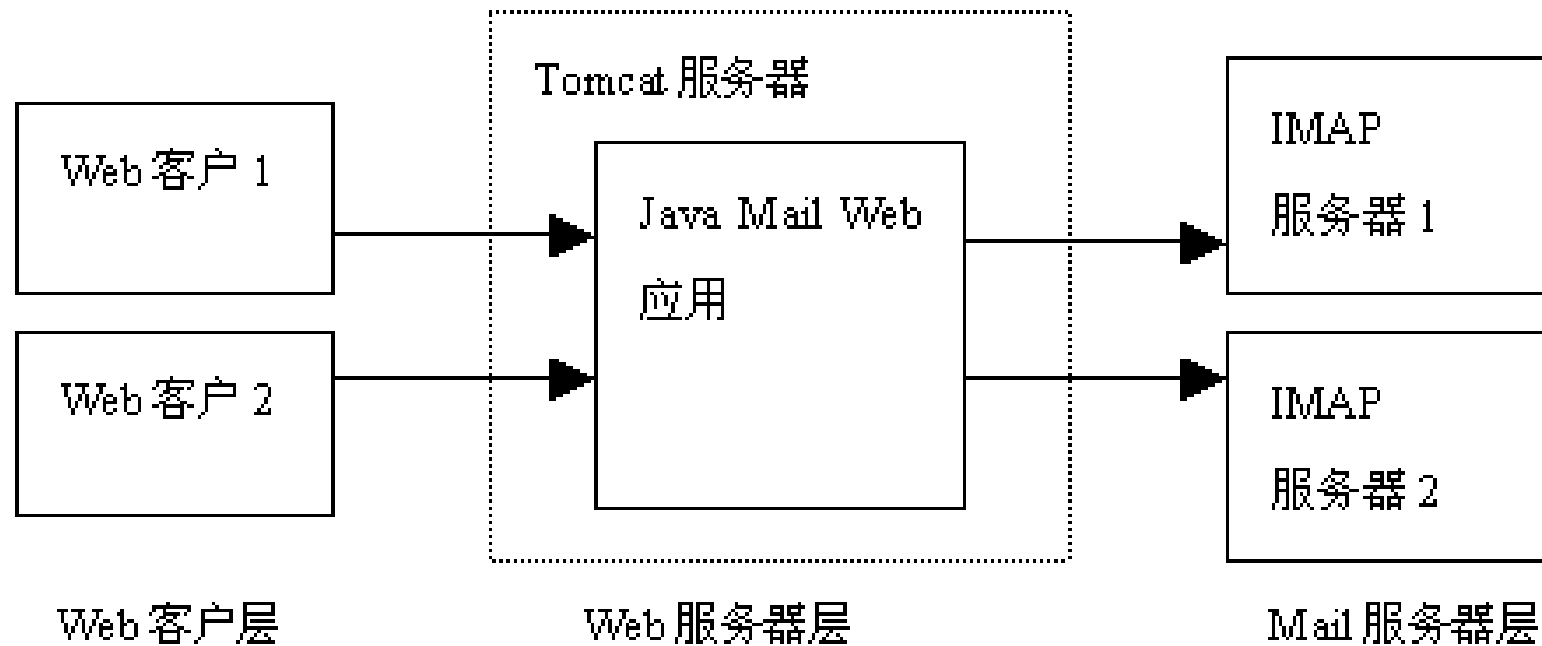
访问邮件服务器上的特定邮件账号

n 从邮件夹中读取信件:

参见MailTest.java

```
//read first Message in inbox  
Message msg=folder.getMessage(1);  
System.out.println("-----the first message in inbox-----");  
System.out.println("From:"+msg.getFrom()[0]);  
System.out.println("Subject:"+msg.getSubject());  
System.out.println("Text:"+msg.getContent());
```

Java Mail Web应用的分层结构





Java Mail Web应用的功能

- n 登录IMAP服务器上的邮件账号
- n 管理邮件夹
- n 查看邮件夹中的邮件信息
- n 查看邮件内容
- n 创建和发送邮件
- n 退出邮件系统

登录IMAP服务器上的邮件账号 (login.jsp)



JavaMail - Microsoft Internet Explorer

文件(E) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退 前进 停止 刷新 搜索 收藏夹 媒体 打印 转送到 链接

地址(D) http://localhost:8080/javamail/login.jsp

Welcome to Java Mail Web

IMAP Mail Server: 127.0.0.1

Username: admin

Password: ****

Login Reset

完毕 本地 Intranet

管理邮件夹 (listallfolders.jsp)

地址(D) <http://localhost:8080/javamail/listallfolders.jsp> 转到 链接 »

Folder List

[CheckMail](#) [Folders](#) [Compose](#) [Logout](#)

FolderName	Total Messages	Unread Messages
inbox	2	1
Trash	32	22
SendBox	11	9
Draft	2	1
test2	1	0

Folder Operation

select operation:

select folder:

new folder name:

查看邮件夹中的邮件信息 (listonefolder.jsp)

地址(D) <http://localhost:8080/javamail/li> 转到 链接 »

CurrentFolder:inbox

[CheckMail](#) [Folders](#) [Compose](#) [Logout](#)

Total Messages:2 Unread Messages:1

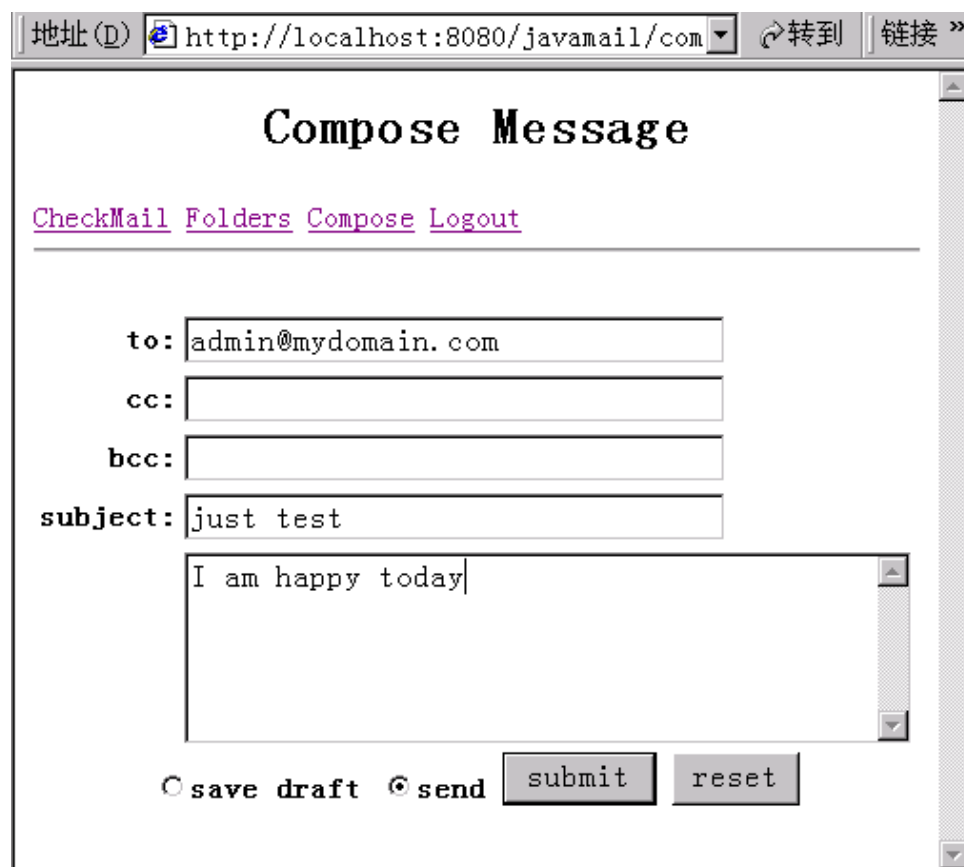
	Sender	Date	Subject	Size
<input type="checkbox"/>	admin	03.09.13 at 11:24:08	hello	394Bytes
<input type="checkbox"/>	admin	03.09.13 at 19:25:44	just test	402Bytes

delete messages

查看邮件内容(showmessage.jsp)



创建和发送邮件（compose.jsp）



地址(D) <http://localhost:8080/javamail/com> 转到 链接 »

Compose Message

[CheckMail](#) [Folders](#) [Compose](#) [Logout](#)

to:

cc:

bcc:

subject:

☐ save draft ☒ send