

第 12 章 PHP 与邮件系统

在实际应用中，往往经常需要使用邮件系统来与网站的访问者交流。例如，一个网站的新注册用户可以通过邮件来获取注册确认信息，也可以通过邮件来找回密码等等。因此，在一些常见的网站系统中，往往包含邮件发送的功能。本章将介绍如何使用 PHP 来发送邮件。

12.1 PHP 邮件发送功能配置

要使用 PHP 发送邮件，首先需要配置 SMTP 服务器。SMTP 是“简单邮件传输协议（Simple Mail Transfer Protocol）”的缩写，是一种可靠并且有效的电子邮件传输协议。使用 PHP 发送邮件的原理也正是通过 SMTP 服务器将 E-mail 发送到指定地址。本节将简要介绍如何配置 PHP 中的邮件发送功能。

12.1.1 PHP 中的 SMTP 功能配置

打开 PHP 安装目录下的 `php.ini` 文件，可以看到如下所示的配置信息。

```
[mail function]
; For Win32 only.
SMTP = localhost
smtp_port = 25

; For Win32 only.
;sendmail_from = me@example.com

; For Unix only.  You may supply arguments as well (default: "sendmail -t -i").
;sendmail_path = "C:\Program Files\xampp\sendmail\sendmail.exe -t"

; Force the addition of the specified parameters to be passed as extra parameters
; to the sendmail binary. These parameters will always replace the value of
; the 5th parameter to mail(), even in safe mode.
;mail.force_extra_parameters =
```

可以看到这里需要配置在发送邮件时使用的 SMTP 服务器地址和 SMTP 服务器所用的端口，以及邮件发送方的地址。读者可以根据实际情况在这里配置自己的 SMTP 服务器信息，然后重新启动 Apache 服务器即可。如果尚没有自己的 SMTP 服务器，下一节将介绍如何快捷的在自己的电脑上配置一个简单的 SMTP 服务器。

12.1.2 建立自己的 SMTP 服务器

目前市场上流行很多 SMTP 服务器软件。例如，Microsoft 的 IIS 就提供 SMTP 服务器的功能。本节要介绍的是一款简单的 SMTP 服务器软件——Mail Direct Pro，该软件可以通过简单的配置使本地机器

具有 SMTP 服务的功能。该软件可以从其官方网站 <http://www.ocloudsoft.com/>处下载到，目前的最新版本是 2.3。安装及配置步骤如下所示。

(1) 解压缩下载后的文件，运行 Setup.exe 文件，启动 PostgreSQL 的安装程序。

(2) 在【Welcome to the Mail Direct Pro Setup Wizard】对话框中单击【Next】按钮，弹出【License Agreement】对话框。

(3) 阅读并接受协议后，单击【Next】按钮，弹出【Select Destination Location】对话框，如图 12-1 所示。

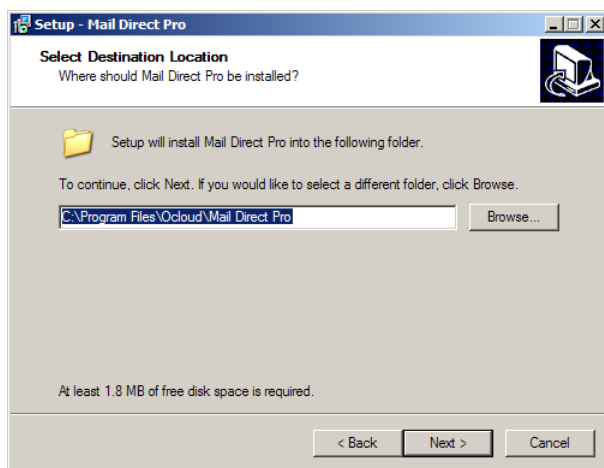


图 12-1 选择目标路径

(4) 选择好安装目标文件夹后，单击【Next】按钮，弹出【Select Start Menu Folder】对话框。

(5) 单击【Next】按钮，弹出【Select Additional Tasks】对话框。

(6) 选择是否需要在桌面上放置图标后，单击【Next】按钮，弹出【Ready to Install】对话框。

(7) 单击【Install】按钮开始安装。

(8) 安装后，选中“Launch Mail Direct Pro”选项，单击【Finish】按钮结束安装。

(9) 在【Mail Direct Configuration Wizard】对话框中，单击【Next】按钮，弹出【DNS Server Configuration】对话框，如图 12-2 所示。

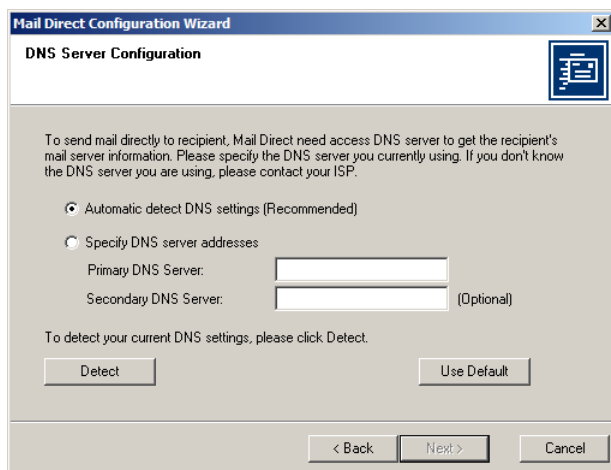


图 12-2 DNS 设置

(10) 根据服务器的实际情况配置 DNS 信息，一般选择【Automatic detect DNS settings】选项，Mail Direct Pro 将自行检测 DNS 设置。也可以通过单击【Detect】按钮，来检测当前的 DNS 设置。设置好以

后, 单击【Next】按钮, 弹出【Change the account settings of your mail client program】对话框, 如图 12-3 所示。

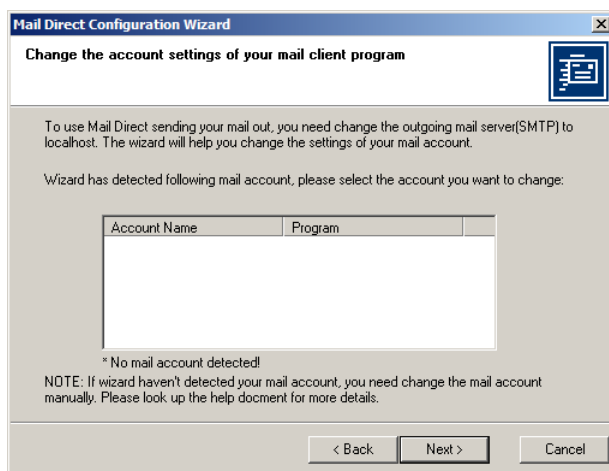


图 12-3 选择要更新的 mail 客户端

(11) 如果在当前服务器上存在用于发送 E-mail 的客户端程序, 可以在此处选择相应的客户端程序, Mail Direct Pro 将更新其 SMTP 服务器配置为 localhost。设置好以后, 单击【Next】按钮, 弹出【Add your current SMTP server to Relay SMTP Server List】对话框, 如图 12-4 所示。

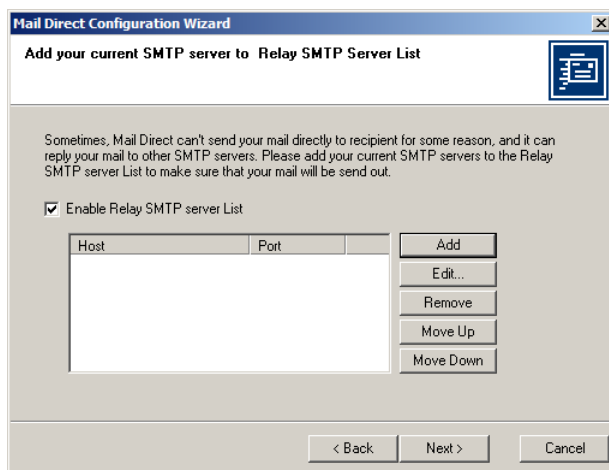


图 12-4 添加 SMTP 服务器

(12) 这里, 可以根据实际情况, 设置一个或多个 SMTP 服务器以确保信件的发出。单击【Add】按钮, 弹出【SMTP Server Information】对话框, 如图 12-5 所示。

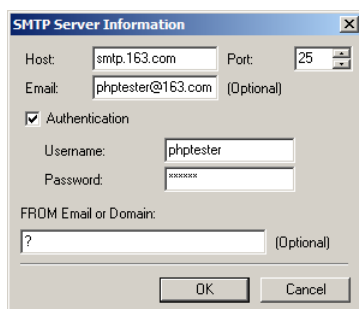


图 12-5 SMTP 服务器信息

(13) 在对话框上填写好服务器的地址和用户名、密码，单击【OK】按钮完成设置。

(14) 设置好以后，单击【Change the account settings of your mail client program】对话框上的【Next】按钮，弹出【General Settings】对话框。

(15) 选择好是否要在 Windows 启动时启动 Mail Direct Pro 选项后，单击【Next】按钮，弹出【Complete the Mail Direct Configuration Wizard】对话框。

(16) 单击【Finish】按钮结束配置，弹出 Mail Direct Pro 主面板，如图 12-6 所示。

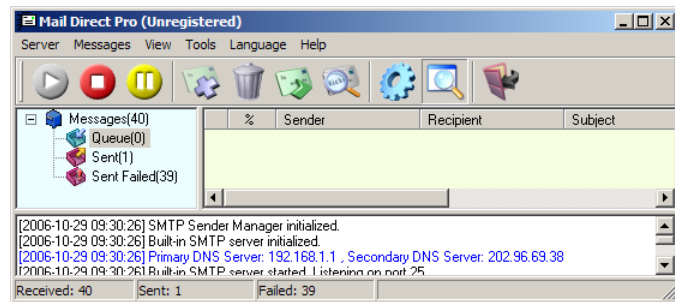


图 12-6 Mail Direct Pro 主面板

如果需要重新配置 SMTP 服务器，可以通过单击【Tool】|【Configuration Wizard】菜单重新启动配置向导。

12.2 PHP 中的 mail 函数

PHP 提供了一个功能强大的 mail 函数用于邮件的发送，在 PHP 中，可以通过 mail 函数来实现各种类型的邮件发送。本节将详细介绍该函数的用法。

12.2.1 简单的邮件发送方法

mail 函数的语法格式如下所示。

```
mail($to, $subject, $body, [$header])
```

其中，\$to 是收件人的 E-mail 地址，如果存在多个收件人，则将地址用逗号“,”分开。\$subject 是 E-mail 的标题。\$body 是邮件的主体部分。\$header 是一个可选的选项，用于表示邮件的头信息。邮件的头信息用于标记邮件的基本信息。如下所示为一个邮件的头信息。

MIME-Version: 1.0	//邮件协议版本信息
Content-type: text/html; charset=iso-8859-1	//数据类型和字符集
To: Simon <simon@php.net>, Elaine <elaine@php.net>	//收件人地址
From: PHP Tester <phptester@163.com>	//发件人地址
Cc: Peter@php.net	//抄送地址
Bcc: David@php.net	//暗抄地址

这里声明邮件使用 MIME 的版本为 1.0，MIME 是“多功能网际邮件扩充协议”的缩写。如果一个邮件使用 MIME 保存，在显示给用户前，浏览器必须根据 MIME 的类型使用相应的应用程序来对邮件进行处理，其中 MIME 的类型使用 Content-type 来表示。这里 MIME 类型为 text/html，表示要传输的是 HTML 文本。除此之外，这里的头信息还声明了字符集 charset 是用 iso-8859-1，收件人信息、发件人信

息、转抄信息、暗抄信息等。以下代码是一个简单的邮件发送例子。

```
<?php
mail("phptester@163.com", "this is a test", "testing");
?>
```

这里的收件人为 phptester@163.com，邮件主题为“this is a test”，邮件的主体为“testing”。这里由于没有指定发件人信息，系统将使用 php.ini 文件中的配置。以下代码指定了发件人信息。

```
<?php
mail("phptester@163.com", "this is a test", "testing",
    "From: simon@php.net\r\n");
?>
```

这里，指定发件人地址为“simon@php.net”。需要注意的是，实际的邮件发送仍然使用 php.ini 上的配置进行发送，但是会将邮件中的发件人信息更改为“simon@php.net”。这样做的效果是收件人看到的邮件是从 simon@php.net 发出的，但是，实际上该地址并不是邮件发出的真正地址。

12.2.2 发送 HTML 格式文本

上一小节中介绍了 MIME 的定义，在邮件的头信息中指定 MIME 类型为 HTML 后，可以使邮件的内容按照 HTML 格式发送，如下面的例子所示。

```
<?php
$to = "simon@php.net, elaine@php.net";           //设置收件人
$subject = "This is a test";                     //设置 E-mail 主题
$message = "                                     //设置 E-mail 内容
<html>
    <head>
        <title>This is a test</title>
    </head>
    <body>
        <p>Dear all,</p>
        <p>This is a test for HTML mail sending.</p>
    </body>
</html>
";

//设置邮件头 Content-type header
$header = "MIME-Version: 1.0\r\n";               //设置 MIME 版本
$header .= "Content-type: text/html; charset=iso-8859-1\r\n"; //设置内容类型和字符集
//设置收件人、发件人信息
$header .= "To: Simon <simon@php.net>, Elaine <elaine@php.net>\r\n"; //设置收件人
$header .= "From: PHP Tester <phptester@163.com>\r\n";           //设置发件人
$header .= "Cc: Peter@php.net\r\n";                             //设置抄送
$header .= "Bcc: David@php.net\r\n";                           //设置暗抄
echo $header;
mail($to, $subject, $message, $header);          //发送邮件
?>
```

这里，在头信息中指定 MIME 类型为 text/html，收件人收到 E-mail 后看到的就是经过 HTML 处理器处理过的邮件信息，如下所示。

```
Dear all,
```

```
This is a test for HTML mail sending.
```

12.2.3 发送带附件的邮件

除了 HTML 邮件可以通过这种形式发送以外，其他的文件也可以通过这种形式发出。如果一个 E-mail 包含多种 MIME 类型，例如，既有图片又有文本，可以在头信息中指定一个 MIME 类型分界线，然后在邮件的主体部分声明 MIME 类型，并用分界线隔开。

以下代码在头信息中构建了一个分界线。

```
$header = "Content-type: multipart/mixed; boundary= 345894369383 \r\n";
```

这里的 boundary 就是分界线，分界线是一串无规律的字符。定义分界线时，MIME 类型指定为 multipart/mixed。

以下代码实现了带附件邮件的发送，用户首先通过浏览器中的表单选择要上传的文件，然后 PHP 代码对上传的文件进行分析和读取，最后将文件混合到邮件主体中发送。

静态 HTML 表单代码如下所示。

```
<html>
  <head>
    <title>带附件的邮件发送</title>
  </head>
  <body>
    <form method="post" name="form1" action="sendmail.php" ENCTYPE="multipart/form-data">
      <table>
        <tr>
          <td>发送人: </td>
          <td><input type="text" name="from"></td>
        </tr>
        <tr>
          <td>收件人: </td>
          <td><input type="text" name="to"></td>
        </tr>
        <tr>
          <td>邮件主题: </td>
          <td><input type="text" name="subject"></td>
        </tr>
        <tr>
          <td>邮件内容: </td>
          <td><textarea name="body"></textarea></td>
        </tr>
        <tr>
          <td>附件上传: </td>
          <td><input type="file" name="upload_file"></td>
        </tr>
        <tr>
          <td colspan="2">
            <input type="submit" value="提交">
            <input type="reset" value="重置">
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```

        </table>
    </form>
</body>
</html>

```

运行结果如图 12-7 所示。



图 12-7 邮件发送界面

用于接收表单的 PHP 代码如下所示。

```

<?php
//获得表单信息
$from = $_POST['from'];
$to = $_POST['to'];
$subject = $_POST['subject'];
$body = $_POST['body'];
// 定义分界线
$boundary = "345894369383"; //分界线是一串无规律的字符
//设置 header
$header = "Content-type: multipart/mixed; boundary= $boundary\r\n";
$header .= "From:$from\r\n";
//获得上传文件的文件内容
$file = $_FILES['upload_file']['tmp_name'];
//确定上传文件的 MIME 类型
$mimeType = $_FILES['upload_file']['type'];
//获得上传文件的文件名
$fileName = $_FILES['upload_file']['name'];
//读取上传文件
$fp = fopen($file, "r"); //打开文件
$read = fread($fp, filesize($file)); //读入文件
$read = base64_encode($read); //base64 编码
$read = chunk_split($read); //切割字符串
//建立邮件的主体，分为邮件内容和附件内容两部分
$body = "--$boundary
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: 8bit
$body
--$boundary
Content-type: $mimeType; name=$fileName

```

```
Content-disposition: attachment; filename=$fileName
Content-transfer-encoding: base64
$read
--$boundary--;
//发送邮件 并输出是否发送成功的信息
if(mail($to, $subject,$body,$header))
{
    echo "信件发送成功";
}
else
{
    echo "信件发送失败";
}
?>
```

如果在代码中输出 \$body 变量，可以看到该变量中已经包含了附件文件的信息。在邮件发送过程中，附件文件将和邮件主体一同发送到目标地址。

12.3 一个简单的邮件发送类

前面介绍了使用 mail 方法发送邮件的方法。在实际应用中，往往会在项目中多处用到邮件发送的功能。如果每次都直接调用 mail 函数来进行邮件发送，不利于后期的维护。因此，往往根据实际需要将邮件发送功能封装成一个类，方便项目的开发和维护。

12.3.1 邮件发送类的创建

本节将介绍一个简单的邮件发送类，实现 HTML 文本和附件的发送，代码如下所示。

```
<?php
class PHPMailer
{
    var $boundary = "345894369383";           //定义分界线
    var $header = "";                          //定义 header
    var $body = "";                           //定义邮件内容
    var $maillist = "";                       //定义收件人列表

    //构造函数
    function PHPMailer()
    {
    }

    //建立 header
    function BuildHeader($from, $to, $cc="", $bcc="")
    {
        $this->header = "Content-type: multipart/mixed; boundary= $this->boundary\r\n";
        $this->header .= "To: $to\r\n";
        $this->header .= "From: $from\r\n";
        $this->header .= "Cc: $cc\r\n";
    }
}
```



```
$this->header .= "Bcc: $bcc\r\n";
}

//建立发送列表
function BuildMailList($to, $cc="", $bcc="")
{
    if($to!="")
    {
        $this->maillist .= $to;
    }
    if($cc!="")
    {
        $this->maillist .= ", ";
        $this->maillist .= $cc;
    }
    if($bcc!="")
    {
        $this->maillist .= ", ";
        $this->maillist .= $bcc;
    }
}

//建立要发送的邮件主体
function BuildBody($htmltext="", $attachment=null)
{
    //将 HTML 文本放入邮件中
    if($htmltext!="")
    {
        $this->body .= "--$boundary
Content-type: text/html; charset=iso-8859-1
Content-transfer-encoding: 8bit
$htmltext";
    }
    //读取附件
    if($attachment!=null)
    {
        if($htmltext!="")
        {
            $this->body .= "--$this->boundary";
        }
        $file = $attachment['tmp_name'];
        //确定上传文件的 MIME 类型
        $mimeType = $attachment['type'];
        //获得上传文件的文件名
        $fileName = $attachment['name'];
        //读取上传文件
        $fp = fopen($file, "r"); //打开文件
        $read = fread($fp, filesize($file)); //读入文件
        $read = base64_encode($read); //base64 编码
    }
}
```

```

        $read = chunk_split($read);                                //切割字符串
        $body .= "Content-type: $mimeType; name=$fileName
Content-disposition: attachment; filename=$fileName
Content-transfer-encoding: base64
        $read";
    }
    //结束分割线
    $this->body .= "--$this->boundary";
}

//邮件发送
function Send($from, $to, $cc="", $bcc="", $subject="", $htmltext="", $attachment=null)
{
    $this->BuildHeader($from, $to, $cc, $bcc);
    $this->BuildMailList($to, $cc, $bcc);
    $this->BuildBody($htmltext, $attachment);
    //发送邮件 并根据邮件是否发送成功来返回不同的 Bool 值
    if(mail($this->maillist, $subject, $this->body, $this->header))
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
?>

```

上面的类可以通过直接调用 `Send` 方法来实现一个复杂邮件的发送。在实际应用中，可以通过直接调用上面的类实现邮件发送的功能。

12.3.2 邮件发送系统的实现

本小节将创建一个允许用户在 Web 页面上实现邮件发送的例子。该例子首先允许用户在表单上输入邮件内容，提交后，用于接收用户输入的 PHP 代码通过调用上一小节的邮件发送类来发送 E-mail。

前台用户表单代码如下所示。

```

<html>
<head>
<title>mailsend</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<style type="text/css">
<!--
.style1 {
    font-size: 18px;
    font-weight: bold;
}
-->
</style>

```

```

</head>

<body>
  <p align="center" class="style1">E-mail 发送系统
</p>
<form action="send.php" method="post" enctype="multipart/form-data">
  <table width="466" border="1" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td width="114"><div align="right"><strong>发信人: </strong></div></td>
      <td width="272"><input name="from" type="text" id="from" size="40"></td>
    </tr>
    <tr>
      <td><div align="right"><strong>收信人: </strong></div></td>
      <td><input name="to" type="text" id="to" size="40"></td>
    </tr>
    <tr>
      <td><div align="right"><strong>抄送: </strong></div></td>
      <td><input name="cc" type="text" id="cc" size="40"></td>
    </tr>
    <tr>
      <td><div align="right"><strong>暗抄: </strong></div></td>
      <td><input name="bcc" type="text" id="bcc" size="40"></td>
    </tr>
    <tr>
      <td><div align="right"><strong>信件主题: </strong></div></td>
      <td><input name="subject" type="text" id="subject" size="40"></td>
    </tr>
    <tr>
      <td><div align="right"><strong>信件内容: </strong></div></td>
      <td><textarea name="body" cols="40" rows="5" id="body"></textarea></td>
    </tr>
    <tr>
      <td><div align="right"><strong>附件上传: </strong></div></td>
      <td><input name="upload_file" type="file" id="upload_file"></td>
    </tr>
  </table>
  <p align="center">
    <input type="submit" name="Submit" value="Submit">
    <input type="reset" name="Submit2" value="Reset">
  </p>
</form>
</body>
</html>

```

用于处理用户输入的 PHP 代码如下所示。

```

<?php
//包含类文件
include("phpmailsender.php");
//定义属性
$from = $_POST['from'];
$to = $_POST['to'];

```

```
$cc = $_POST['cc'];
$bcc = $_POST['bcc'];
$subject = $_POST['subject'];
$body = $_POST['body'];
$file = $_FILES['upload_file'];
//发送 E-mail
$phpmailsender = new PHPMailer(); //定义 PHPMailer 对象
if($phpmailsender->Send($from, $to, $cc, $bcc, $subject, $body, $file))
{
    echo "<script>alert('信件发送成功');</script>";
}
else
{
    echo "<script>alert('信件发送失败');</script>";
}
?>
```

运行效果如图 12-8 所示。

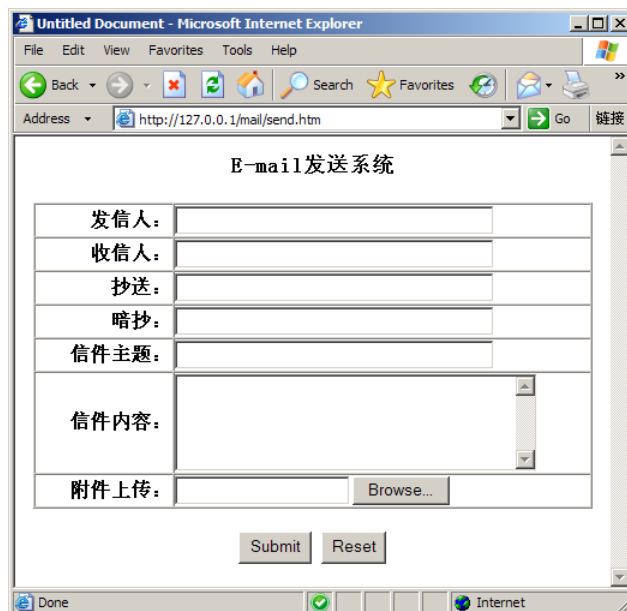


图 12-8 信件发送页面

单击【Submit】按钮后，用户的输入将被 PHP 代码以 E-mail 的形式发送到相应的信箱中。可以看出，由于详细的邮件发送代码都被封装起来，发送一个复杂的邮件也只需要简单的几行代码。这样，即使需要对邮件发送功能进行一些改动，也只需要对类中的代码进行更新即可。

12.4 小结

本章介绍了如何使用 PHP 的 mail 函数进行邮件发送，并提供了一个简单的邮件发送类。邮件发送功能在论坛系统、信息订阅系统等实际应用中使用时非常普遍。在实际应用中，读者可以通过对本章中的邮件发送类进行研究，并根据实际项目的情况编写出更合适的用于邮件发送的类，为整个项目提供邮件

发送支持。